
TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: N2612 – Elektrotechnika a informatika

Studijní obor: 1802T007 – Informační technologie

Optimalizace richformátu webových prezentací

Optimalization Rich Format of web presentation

Diplomová práce

Autor: Bc. Ondřej Kratochvíl

Vedoucí práce: RNDr. Klára Císařová, Ph.D.

V Liberci 20. 5. 2011

Originál zadání

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum

Podpis

Poděkování

Na tomto místě bych rád poděkoval své vedoucí diplomové práce RNDr. Kláře Císařové, Ph.D. za její neocenitelné rady a inspiraci při řešení práce. Dále bych chtěl poděkovat celé rodině a přítelkyni, která mě během studia podporovala.

Abstrakt

Diplomová práce představuje publikační e-learningový systém Mediasite, který Technická univerzita v Liberci využívá dnes už jako inovativní součást e-learningových technologií používaných na fakultě mechatroniky, informatiky a mezioborových studií. S provozem Mediasite je spojeno několik problémů, které se na fakultě postupně řeší. Tato práce je příspěvkem k optimální implementaci Mediasite na TUL. Řešení bylo podmíněné studiem systému a také zmapováním stavby programu, který celý systém Mediasite zastřešuje, aby bylo možné do tohoto systému vhodně zasáhnout.

Mediasite umožňuje záznam přednášek ve tvaru richmédií. Ve výsledné webové prezentaci je navigace určena jednotlivými snímky prezentace doprovázejícími přednášku. Tato filozofie vyhovuje pouze omezenému typu přednášky, kdy je počet „slidů“ přiměřený. Přednášky, pro které je typická jiná technika výkladu a u kterých je nutné pořizovat například záznam „z tabule“, má ve výsledném tvaru stovky až tisíce „slidů“. Navigace je tak zcela znehodnocena. Diplomová práce řeší způsob jak redukovat „slidy“ pro navigaci a zároveň zachovat všechny důležité „slidy“ tak, aby výklad přednášeného tématu zůstal neporušený. Základní požadavek byl, aby redukce počtu „slidů“ pro navigaci i úprava výsledného HTML kódu byla automatická. „Ruční“ editace záznamu ve tvaru richmédií v předložené aplikaci je možná, ale potřebné lidské zdroje by byly nepřiměřené současným možnostem fakulty.

Zejména proto poslední částí diplomové práce je testování aplikace a jejich porovnávání metod pro redukci počtu „slidů“ na různých typech záznamů odvozených od typů přednášky. Testováním byla získána představa o nastavení parametrů optimalizace tak, aby automatizace prací mohla být co největší.

Klíčová slova

Mediasite, náhled, richmédiá, vícedruhová média, redukce, e-learning, slide, streamování videa, porovnání obrázků, streamovaná média, publikační systém.

Abstract

Diploma thesis introduces e-learning system for publication of presentations, Mediasite which is used at Technical university of Liberec as a part of innovative e-learning technologies used at Faculty of Mechatronics. There are a few problems with optimal working of Mediasite system which are currently solved at the faculty. This work is a contribution to better implementation of Mediasite at TUL. Solving of the problems is conditional upon studying whole Mediasite system to map its software structure which covers containing hardware for purpose to appropriate intervene with the system. Mediasite brings recording of lessons and publish them as richmedia. In resulting web presentation is navigation created by each slide. This philosophy is only suitable for limited kinds of presentations. Presentations where other technics of interpretation are used such as using the multimedia blackboard or the visualiser, are much worse for students to navigate in. Diploma thesis solves this exact problem with navigation in way how to reduce captured set of slides without losing any context. Main goal was to reduce the slide set and to update the HTML template automatically. Manual adjustments are also available but the human resources would be disproportionate to the current capabilities of the faculty.

These are the reasons why the last part of diploma thesis contains tests of used comparative methods for slide reduction in any kind of presentation. The output was to get an idea about optimal adjusting parameters for reduction so that the application would be fully automatic.

Keywords

Mediasite, preview, Rich media, bitmap reduction, E-learning, slide, streaming video, picture comparison, streamed media, publication system.

Obsah

Prohlášení	3
Poděkování	4
Abstrakt.....	5
Seznam zkratek.....	9
Seznam obrázků.....	10
Seznam grafů	11
Úvod	12
1 E-learning	14
1.1 Elektronický kurz.....	14
1.2 Standardy e-learningu	15
1.3 Řízení elektronických kurzů	16
1.4 Prostředky a podmínky pro e-learning.....	16
2 Richmédia a e-learning	18
2.1 Výhody richmédií	19
3 Řešení na TUL	20
3.1 Mediasite.....	21
3.1.1 Rozhraní Mediasite	22
4 Moderní cesty porovnávání multimédií.....	25
4.1 Systém MUFIN.....	26
4.2 Metoda s využitím OCR	27
4.2.1 Rozpoznávání podle předlohové matice	28
4.2.2 Rozpoznávání podle charakteristických tahů písmen	28
5 Šablona systému Mediasite.....	30
5.1 Definice struktury slidů	30
5.2 Chování seznamu náhledů	31
5.3 Provázání videa a obrázků	32
5.4 Definice systémových událostí Mediasite	32
5.5 Rozšíření systémových událostí.....	32
5.6 Úprava šablony pro redukci slidů	33
5.6.1 Úprava struktury po redukci	33
5.6.2 Úprava chování, zobrazení a akcí	34
5.6.3 Úprava provázání prezentace	34

5.6.4	Úprava systémových událostí a časování videa	34
6	Návrh optimalizované navigace v Mediasite	36
7	Návrh aplikace RichMedia Slider	38
7.1	Uživatelské rozhraní	38
7.2	Metoda průměrného počtu rozdílných pixelů	41
7.3	Porovnání vzdálenosti v RGB prostoru	41
7.4	Průměrná hodnota rozdílu pixelů	42
7.5	Kombinační metoda výpočtu	42
7.6	Redukce obrázků	43
7.7	Použité metody a techniky	44
7.7.1	Gradient na vykreslení formuláře	44
7.7.2	Zobrazení obrázku z náhledové lišty	45
7.7.3	Načítání sady slidů	47
7.7.4	Zobrazení checkboxů v náhledové liště	47
7.7.5	Funkce pro zjištění počtu obrázků	48
7.7.6	Algoritmy porovnávacích metod	49
7.7.7	Redukce sady slidů	50
7.7.8	Překreslení okna při náročných operacích	51
7.7.9	Šetrná práce s pamětí při výpočtech	52
7.8	Test na konkrétních typech přednášek	52
7.8.1	Test PP – přednes a prezentace	54
7.8.2	Test PIT – prezentace IT předmětů	56
7.8.3	Test PVP – přednáška s výkladem pojmů	58
	Závěr	61
	Literatura	63
Příloha A	– Popis přiloženého CD	65
Příloha B	– Návrhové zobrazení optimalizované navigace	66

Seznam zkratek

HTML	HyperText Markup Language
SCORM	Shareable Content Object Reference Model
LMS	Learning Management System
LCMS	Learning Content Management System
MUFIN	MUlti Feature Indexing Network
OCR	Optical character recognition
CRC	Cyclic redundancy check

Seznam obrázků

Obrázek 1: Struktura systému Rich Media	18
Obrázek 2: Úvodní stránka systému Mediasite	23
Obrázek 3: Stránka prezentace v Mediasite.....	23
Obrázek 4: Náhledové okno prezentace	24
Obrázek 5: Rozhraní systému MUFIN	27
Obrázek 6: Maticový seznam klíčových bodů prezentace.....	37
Obrázek 7: Řádkový výpis klíčových bodů prezentací	37
Obrázek 8: Uživatelské rozhraní aplikace	39
Obrázek 9: Nastavení formátu názvu obrázků.....	40
Obrázek 10: Vzdálenost v RGB prostoru krychle	42
Obrázek 11: Provedená redukce	44

Seznam grafů

Graf 1: Rozdílnost slidů prezentace ALG 1. metodou.....	54
Graf 2: Rozdílnost slidů prezentace ALG 3. Metodou	55
Graf 3: Rozdílnost slidů prezentace ALG kombinační metodou.....	55
Graf 4: Rozdílnost slidů prezentace GDA 1. metodou	56
Graf 5: Rozdílnost slidů prezentace GDA 3. metodou	57
Graf 6: Rozdílnost slidů prezentace GDA kombinační metodou	57
Graf 7: Rozdílnost slidů prezentace PAS 1. metodou	58
Graf 8: Rozdílnost slidů prezentace PAS 3. metodou	59
Graf 9: Rozdílnost slidů prezentace PAS kombinační metodou.....	59

Úvod

Na serveru „<http://www.prednasky.tul.cz>“ mohou studenti Technické univerzity v Liberci pohodlně sledovat on-line nebo off-line přednášky již přibližně 2 roky. Praxe ukázala problém s navigací v záznamu. Přednášky jsou nahrávány videokamerou a v průběhu tohoto nahrávání jsou synchronně snímány jednotlivé obrázky (slidy), které vyučující promítá pomocí projektoru. V učebně je nainstalována multimediální tabule, která může být také zdrojem datového proudu zobrazeného přes dataprojektor. V další učebně pro záznam přednášek se podobně používá tzv. mimio anebo vizualizér, o kterých bude řeč v příslušné kapitole. Vše se zpracovává komerčním systémem Mediasite, který indexuje a synchronizuje veškeré výstupy prezentace. Data z dataprojektoru jsou do výsledku zaznamenávána jako obrázky formátu JPEG při každé změně v prezentaci. Při zápisu na tabuli nebo použití vizualizéru pro zápis výkladu to znamená vkládání JPEGů do webové prezentace a to až v počtu čtyřech snímků za sekundu. Formát webových stránek se tím zachová v potřebné kvalitě zobrazení, ale zároveň tento typ výkladu přednášky způsobuje nadměrné generování vlastních snímků ve formátu JPEG. Původní myšlenka, že prezentace přibližně o 40 slidech na přednášku, zároveň slouží jako osnova, či navigace v obsahu přednášené látky se zcela znehodnotí. Používání systému Mediasite na všechny typy přednášek tak překračuje prvotní myšlenku navrženého formátu a vzniká problém, který bylo potřebné vyřešit.

Součástí práce je shrnutí jednotlivých druhů e-learningu a jejich využití v praxi kam použití MediSite patří. V teoretické části bude věnován prostor samotnému systému Mediasite a popisu jeho částí a funkcí. Pro vlastní řešení je podstatným tématem studium tzv. richmédií, která jsou v poslední době stále používanější právě ve spojení s e-learningem a s deklarovaným problémem.

Hlavní částí práce je vývoj aplikace, která bude schopna z generovaných snímků vyřadit irelevantní snímky nebo snímky, které jsou znásobeny například technickou poruchou – chvění obrazu apod. Cíle je vyřadit irelevantní snímky z navigační osnovy webové stránky a výkladovou část v datovém okně zachovat v nezměněné formě. Zamýšlená aplikace by měla umožňovat plně automatický přístup za účelem zjednodušení současného post-processingu. V dalším kroku by pak aplikace zpracovala HTML šablonu, která je za zobrazování publikované prezentace zodpovědná a upravila

ji do formátu, kdy se na hlavní straně zobrazují veškeré snímky prezentace a v menu si poté student může vybírat mezi zásadními body prezentace. Dojde tedy k optimalizaci navigace v prezentacích, která přinese přívětivější ovládání pro studenty. Pro zpracování HTML šablony bude nejprve nutné podrobně nastudovat celou strukturu programové části prezentací v systému Mediasite. Běh videa, střídání slidů a multimediální chování zajišťuje několik skriptů v jazyce *JavaScript*, se kterými je taktéž nutné se seznámit. Neméně důležitým úkolem je navrhnout zamýšlenou aplikaci tak, aby vyžadovala minimální uživatelský vstup a vykazovala co nejnižší zatížení počítače.

V další části bude navržena změna navigační části systému Mediasite, kde bude docíleno vhodného provázání systému s aplikací, která z této diplomové práce vychází. Hlavní myšlenkou je zachování plné palety původních slidů v prezentacích vedle výsledné navigační sady.

Nakonec bude věnován prostor testům, kterým bude aplikace podrobena. Bude nutné ověřit funkčnost aplikace a také způsoby, jakými se program vyrovná s krajními případy prezentací, které obsahují mnoho nasnímaných obrázků, nebo naopak obsahují méně obrázků, které jsou si podobné nebo totožné. Výsledky budou prezentovány pomocí grafů, které vyhodnotí funkčnost aplikace v praxi.

1 E-learning

E-learning je poměrně nová metoda výuky, která se postupem času dostává čím dál více do popředí. Předchůdcem e-learningu byly takzvané vyučovací elektronkové strojky a také vzdělávací software nasazený na sálové počítače. V té době bylo využití e-learningu velice ekonomicky nákladné, což do jisté míry platí dodnes. Největší impuls k rozvoji e-learningu přišel v době rozšíření internetu. Tehdy se ještě používal starší název „online learning“.

V současné době se internetové technologie rychle vylepšují a díky tomu dochází k pokroku i v oblasti e-learningu. S vývojem e-learningu je ovšem nutné, aby zároveň realizátoři e-kurzů měli dostatečné povědomí o možnostech výuky a využívali moderní technologie.

E-learning je možno chápat v několika rovinách. V první řadě se jedná o systém, který využívá počítačovou techniku k šíření vzdělávacího obsahu nejčastěji prostřednictvím internetu. Dále je možné chápat e-learning také jako formu výuky při využití multimédií. V neposlední řadě je možné říci, že e-learning je proces komunikace mezi studentem a pedagogem.

Výuka prostřednictvím PC za pomoci nejmodernějších technologií má za cíl co nejvyšší integraci do běžné výuky. Důvodem je hlavně okamžitá a jednoduchá přístupnost materiálů pro studenta. E-learning se často využívá v distanční formě studia, kdy je čas strávený na půdě školy minimalizován na co nejkratší dobu. V tomto případě je získání elektronického materiálu výbornou možností pro samostudium studentů. Může se jednat o přednášky nebo audio a video záznamy. E-learning se využívá nejen ve školství, ale také v rámci celoživotního vzdělávání uvnitř firem.

V poslední době se na mnoha školách prosazuje forma blended learning¹, kde se nasazují moderní e-learningové metody pro doplnění prezenční formy studia. Pojem blended learning zahrnuje také využívání výukových CD-ROM a dalších pomůcek.

1.1 Elektronický kurz

Elektronický kurz je základním kamenem e-learningu. Většinou obsahuje více předmětů a pokrývá tak většinu požadovaných znalostí. Z pohledu programátora se

¹ Blended learning – kombinace standardní výuky s e-learningem

nejedná o nic jiného než soubor obsahující informace k určitému oboru vědomostí. Z hlediska pedagoga se jedná o mimořádně užitečný zdroj informací pro studenty a také zajímavou formu prezentace dané látky, ale podstatné je také to, že to je ověřený zdroj informací.

Důležitou složkou pro vytvoření kvalitních elektronických kurzů je i dobrý software, pomocí kterého se informace prezentují. Díky takovému softwaru je možné zadávat studentovi úlohu, zpřístupnit výuku a tak zvýšit pravděpodobnost pochopení daného problému. Dalším kladem je také možnost zpětné vazby, která je ve výuce velmi důležitá.

Možnost vytvářet elektronické kurzy předpokládá, že pedagog ovládá určité autorské nástroje. Důraz je kladen na dvě základní rozdělení obsahu do tzv. znovupoužitelných objektů (reusable learning objects) a znalost standardů, které se elektronických kurzů týkají. Znovupoužitelnými objekty je možné chápat například digitální obrázky, části různých textů, animace, ale jedná se i o celé webové stránky, díky kterým se informace rozšiřuje jednodušší formou. Hlavní myšlenkou je, aby byly tyto objekty opakovaně využity v mnoha vzdělávacích kurzech.

1.2 Standardy e-learningu

Standardy, které se váží k elektronickým kurzům, zajišťují přenositelnost informací mezi softwarovými řešeními různých firem. Díky těmto standardům je také zjednodušená komunikace mezi kurzy a řízením kurzů. O vytváření standardů se stará například IMS Global Learning Consortium Inc., W3C² nebo IEEE³. Základními standardy, které se využívají, jsou například HTML nebo XHTML formáty, kterými lze snadno zobrazovat obsah kurzů přímo na internetu. Díky tomu jsou tyto formáty hojně využívány firmami, které vyvíjí e-learningové softwary. V dnešní době stojí v popředí spíše formát XHTML, který využívá výhody značkovacího jazyka XML. Jeden z prvních standardů, jímž se e-learning řídil je standard AICC⁴. Tento standard koriguje, způsob výměny mezi výukovými kurzy a systémy. Dalším standardem je IMS, který popisuje výměnu dat pomocí již zmíněného XML, ale také popisuje mimo jiné metodickou kvalitu výuky.

²The World Wide Web Consortium

³Institute of Electrical and Electronics Engineers

⁴Aviation Industry CBT Committee

Nejdůležitějším prvkem v oblasti standardů je SCORM (Sharable Content Object Reference Model). Jedná se o soubor doporučených metod, které by se měly při vytváření e-learningových nástrojů dodržet. SCORM byl vytvořen i kvůli propojení akademických a průmyslových standardizačních konsorcií. Tento model zajišťuje vícerozměrnou interoperabilitu a taktéž využívá značkovacího jazyka XML. Momentálně existuje ve verzi SCORM 2004.

1.3 Řízení elektronických kurzů

Řízení elektronických kurzů v prostředí internetu zabezpečuje software pro řízení studia LMS (Learning Management System). LMS je virtuální prostředí, díky kterému je student schopen jednoduše vyhledávat v poskytnutých informacích nebo komunikovat s pedagogem. Částmi LMS bývají často diskusní fóra nebo jiné komunikační nástroje. Díky LMS může pedagog mimo jiné monitorovat počet spuštění určitého obsahu a tak získávat statistické údaje. S LMS je často spojován termín LCMS (Learning Content Management System). Jedná se o nástroj pro tvorbu nebo testování výukového obsahu e-learningu. LCMS si lze představit jako systém, který úzce spolupracuje s LMS a jejich propojení zajišťuje standard SCORM.

1.4 Prostředky a podmínky pro e-learning

Prostředky, bez kterých se žádný e-learningový nástroj neobejde, mohou být:

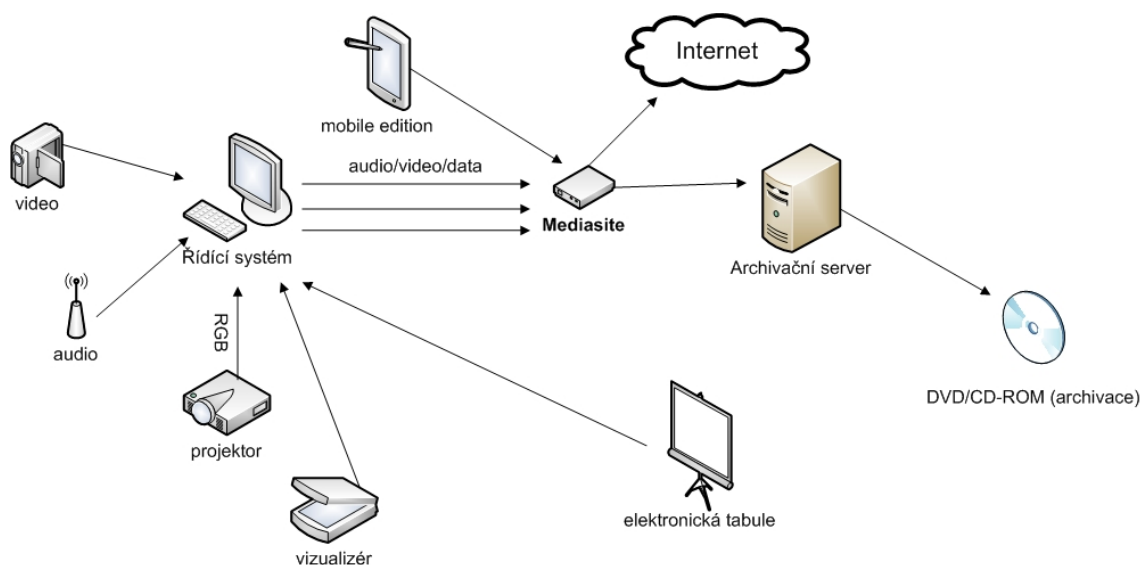
- stolní počítač pro přístup k multimédiím a internetu,
- softwarové prostředky – nástroje, pomocí kterých dochází k vytváření kurzů,
- autorské nástroje – nástroje, které definují podmínky k absolvování kurzů a vytváří jeho strukturu.

Podmínkou k úspěšnému nasazení e-learningu je hlavně zájem studentů, pro který je takový systém zamýšlen. Zájem studentů by měl motivovat i pedagog. Další podmínkou jsou kvalitní a přístupné kurzy, které jsou pro studenta srozumitelné. Jednou z podmínek je také dostatečný počet míst, odkud je možné do elektronických kurzů přistupovat. Je nutné zajistit pro studenty dostatečný počet počítačů. Poslední podmínkou je alespoň minimální uživatelská znalost práce na PC, bez které není možné elektronické kurzy studovat.

Předpokladem ke správnému fungování e-learningu je vhodná a pravidelná distribuce znalostí, které by měly být přístupné pro všechny uživatele.

2 Richmédia a e-learning

Systémy podporující richmédia jsou v dnešní době velmi často využity k podpoře e-learningu. Jedná se o paralelní prezentaci záznamů více druhů datových proudů, odtud také název (Rich Media – bohatá, vícedruhovná média). Nejčastěji jde o audio a video záznamy nebo záznamy prostřednictvím multimediální, elektronické tabule, vizualizéru apod. Záznamy je možné prezentovat na webu nebo archivovat. Richmédia obsahují zvukový, obrazový a digitální obsah všeho druhu. Nákres jak richmédia fungují je vidět na obrázku 1.



Obrázek 1: Struktura systému Rich Media

V praxi to znamená, že záznam richmédií musí být řízen řídicím systémem podle technického vybavení pracoviště. V pedagogické praxi při každodenní potřebě pořizování záznamu došlo k nutné automatizaci a tím ulehčení manipulace se zařízeními používanými na daném místě. Pedagog tak stlačením jediného tlačítka může spustit záznam výkladu, konference nebo jiného setkání. V průběhu záznamu se jednotlivá média převádějí a synchronizují do tvaru, který je dále editovatelný. Pro publikaci a následnou archivaci se často záznam stříhá, ale lze upravovat i jednotlivé datové či zvukové části výsledné synchronní prezentace. Po úpravách jsou všechny výsledky uloženy na server, který informace katalogizuje a soubory přehledně organizuje. Pro záznam richmédií je možné využít i mobilní verzi systému, která je vybavena LCD displejem a objemným diskem pro záznam kongresů, výstav a dalších akcí.

2.1 Výhody rich médií

Neocenitelnou výhodou je vyšší kvalita „vícedruhových“ médií pro vzdělávací proces než je u klasických streamů. K paralelnímu propojení většího počtu datových proudů došlo právě z důvodu zvýšení informačního obsahu záznamů. Video záznam, který se používal v e-learningu v minulosti, nebyl ideální z důvodu špatné rozlišovací schopnosti obrazu a horších audio stop. Menší texty popřípadě nákresy na tabulích nebyly pro studenta takřka vůbec čitelné. Pro akademické prostředí je důležitá právě dobrá audio kvalita, ale i dobrá rozlišovací schopnost a to i na úkor nižší frekvence snímků. Řešením problémů se špatnou kvalitou zvuku může být v tomto případě tzv. klopový mikrofon, který odstraní zpětnou vazbu a další šum. Pokud je požadována volnost přednášejícího, může být zvolena také bezdrátová zvuková sada vybavená bezdrátovým mikro-portem.

Pro kvalitní záznam „výkladu na tabuli“ lze využít datového proudu rich médií a technicky jej získávat z různých typů zařízení. Běžně se používá vizualizér, který má výhodu také v tom, že pedagog neztrácí vizuální kontakt se studenty. Problém je pouze v jeho méně obvyklém využití. Mnohem častěji se využívá tzv. mimio. Jedná se o technologii firmy Virtual Ink Corporation (www.mimio.com), která vytvoří z běžné bílé magnetické tabule tabuli elektronickou. Patentovaná „Stylus Tracking Technology“ umožňuje pomocí přenosného snímače (IR záření + ultrazvuk) určit přesnou polohu čtyř barevných popisovačů, které jsou vloženy do pouzder s vestavěnými vysílači. Obsah psaného projevu přednášejícího je trvale ukládán v reálném čase do počítače a lze jej kdykoliv vytisknout, uložit ve formátech BMP, JPEG, nebo exportovat ve formátu HTML či jej promítnout přes dataprojektor a tím ho dostat do rich médií v příslušné části webové stránky.

3 Řešení na TUL

Technická univerzita v Liberci, po několika měsících testování, uvedla úspěšně do provozu záznam přednášek a konferencí hned v několika učebnách. Studenti mají takřka ihned přístup ke streamovaným přednáškám včetně všech náčrtů na elektronickou tabuli nebo vlastních snímků prezentace pedagoga.

V první etapě úvahy o využití richmédií na TUL se přemýšlelo o vývoji vlastní technologie. Nakonec byla zvolena cesta komerčního řešení Mediasite od společnosti Sonic Foundry, která poskytovala veškerý uživatelský komfort a splňovala všechny nároky, které byly na systém kladeny. Systém Mediasite poskytuje přehledné rozhraní, kde na jednom místě může student prohlížet streamované video, snímky prezentací a poslouchat výklad, který k přednášce patří. Student může také listovat mezi jednotlivými snímky prezentace a přesouvat se v přednáškách na jakýkoliv časový index. Dále si v menu může vybrat zobrazení zmenšených snímků s náhledy a výpis jednotlivých kapitol prezentace.

Součástí systému Mediasite je i projektor, který posílá aktuálně zobrazené snímky do databáze pro budoucí prohlížení. Projektor odešle snímek okamžitě po změně zobrazovaného obrazu. Tím se vytvoří nový bod, na který je možné přejít z navigačního panelu aplikace. Díky tomu, že například při praktické prezentaci určitého software dochází k časté změně obrazu, se vytvoří stovky bodů do navigačního menu. Tato práce se zabývá mimo jiné rozbořením právě tohoto problému a možnostmi jeho řešení. Výsledná aplikace by měla podle různých metod a nastavené úrovně redukce ponechat v navigačním menu pouze relevantní body. Nastavení úrovně redukce je nutné k tomu, aby mohly být porovnávány obrázky ze všech typů přednášek. Například v matematických předmětech je nutné ponechat více obrázků z hlediska celistvosti výkladu.

Na univerzitách obecně lze najít několik základních způsobů přednášení [3]:

1. Přednes a prezentace (Typ PP) – ve výsledném záznamu je streamované video synchronizováno s prezentací, kterou dodal pedagog.
2. Přednes a výklad pomocí elektronické tabule – tabule je využívána například k výkladu matematických nebo fyzikálních pojmů (Typ PVP).
3. Přednášky IT – přednášky doplněné praktickými ukázkami práce na počítači, kde je možné ukázat software týkající se výkladu (Typ PIT).

4. Přednášky s video-prezentací – přednášky, které se často využívají v medicíně, architektuře nebo videa mohou dokreslovat různé experimenty (Typ PAV).

Pro definované typy přednášek má fakulta odzkoušené varianty speciálního technického vybavení tak, aby výsledný tvar richmédií byl pro vzdělávání co nejlepší. Optimalizace vybavení vzhledem k typu přednášek ale také rozvrhovým akcím a ochoty pedagogů nechat se nahrávat je však velmi komplikovaná.

3.1 Mediasite

Mediasite je kompletní systém pro záznam a publikaci „vícedruhových“ médií. Jak již bylo zmíněno, systém pracuje plně automatizovaně, takže lze celý proces ovládat jedním tlačítkem. Správa a distribuce záznamů pracuje také automaticky.

Systém Mediasite se skládá z Mediasite recorderu a z Mediasite serveru, který zajišťuje rychlou komunikaci s recorderem. Na serveru také probíhá automatická indexace nahraného obsahu. Zároveň lze ze serveru vytvořit CD-ROM se záznamy, které je tak možné dále distribuovat nebo jinak archivovat.

Systém zpracování probíhá tak, že se nejdříve spustí přednáška, poté se spustí záznam audio vstupu a video vstupu. Recorder vše zaznamená, zakóduje a také synchronizuje audio vstup s videem a další vstupní grafikou. Součástí Mediasite je také uživatelské rozhraní, ve kterém lze jednoduše procházet všechny nahrávky. Po úspěšném zaznamenání přednášky lze vygenerovat robustní HTML šablonu stránky, která již obsahuje veškeré zaznamenané snímky s odpovídajícími časovými indexy a také záznam videa prezentace. V případě využití multimediální tabule jsou snímky proložené i náčrtky vyučujícího. Po zveřejnění a spuštění vygenerované šablony již mohou studenti sledovat přednášky na internetu.

Mezi nasnímáním přednášky a vygenerováním HTML šablony je velice důležitý krok post-processingu. Během něj je celý záznam stažen ze serveru, který přednášku nahrál. Poté je možné upravit video případně doplnit chybějící údaje o přednášce např. jméno vyučujícího, název předmětu apod. Tento proces je velice časově náročný a v případě rozsáhlejšího zásahu do prezentace je velice problematický. Po tomto kroku již lze šablonu ze systému vygenerovat a přednášku úspěšně publikovat.

Rozhraní umožňuje nastavení autorských práv, takže je možné konkrétnímu studentovi zpřístupnit jen některé informace a k přednáškám tak mohou mít přístup pouze studenti, který daný kurz navštěvují.

Systém Mediasite již byl úspěšně nasazen v mnoha posluchárnách univerzit po celém světě. V České republice jej využívá Vysoká škola báňská – Technická univerzita Ostrava. Díky tomuto systému si mohou studenti připomenout učivo přednášky kdekoliv, kde mají přístup k internetu. V komerčním prostředí se Mediasite využívá například pro publikaci firemních prezentací nebo důležitých organizačních jednání. Mediasite se také nasazuje do státní správy, kde jsou jeho pomocí sdělovány důležité informace například záznamy z mezinárodních jednání apod.

3.1.1 Rozhraní Mediasite

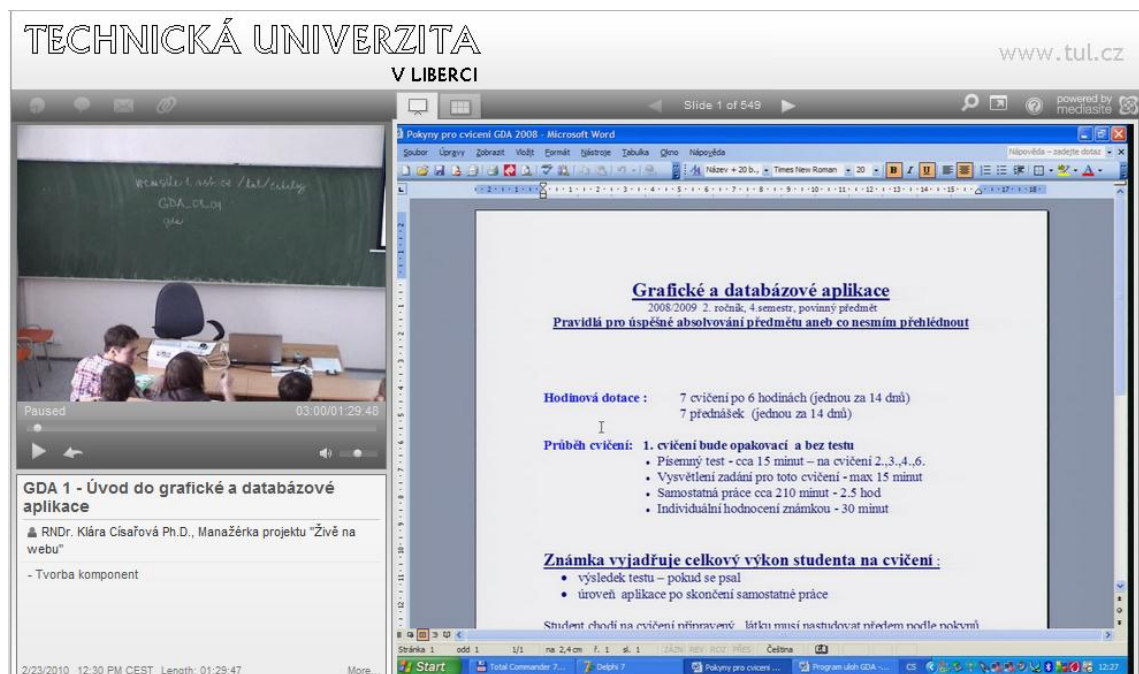
Rozhraní systému Mediasite je navrženo podobně jako běžná internetová fóra nebo jiné e-learningové weby, jako je například Moodle⁵ a další. Na úvodní stránce lze nalézt nejaktuálnější videa, která se týkají celoškolních aktivit. Úvodní stránka obsahuje několik ovládacích prvků (viz Obrázek 2). Po zalogování, kliknutím na tlačítko „Login“ v pravém horním rohu, se uživatel dostane do katalogu jednotlivých předmětů. Tato stránka vypadá totožně jako úvodní a nabízí výběr ze záznamů týkajících se konkrétního předmětu. V levém sloupci je možné procházet stromovou strukturu uložených prezentací a díky tomu vybírat například konkrétní semestr, kdy byl předmět vyučován.

⁵ Moodle - Open Source software pro tvorbu elektronických kurzů



Obrázek 2: Úvodní stránka systému Mediasite

Záznamy se spouštějí poklikáním na název přednášky nebo na její doplňující obrázek. Poté se otevře rozhraní, které zajišťuje přehrávání videa, prohlížení snímků prezentace a listování miniaturami snímků, které je možné zobrazit jako jednoduchý řádkový výčet nebo jako zmenšené náhledové slidy. Zobrazení miniatur slidů se nachází v tzv. náhledovém okně prezentace.

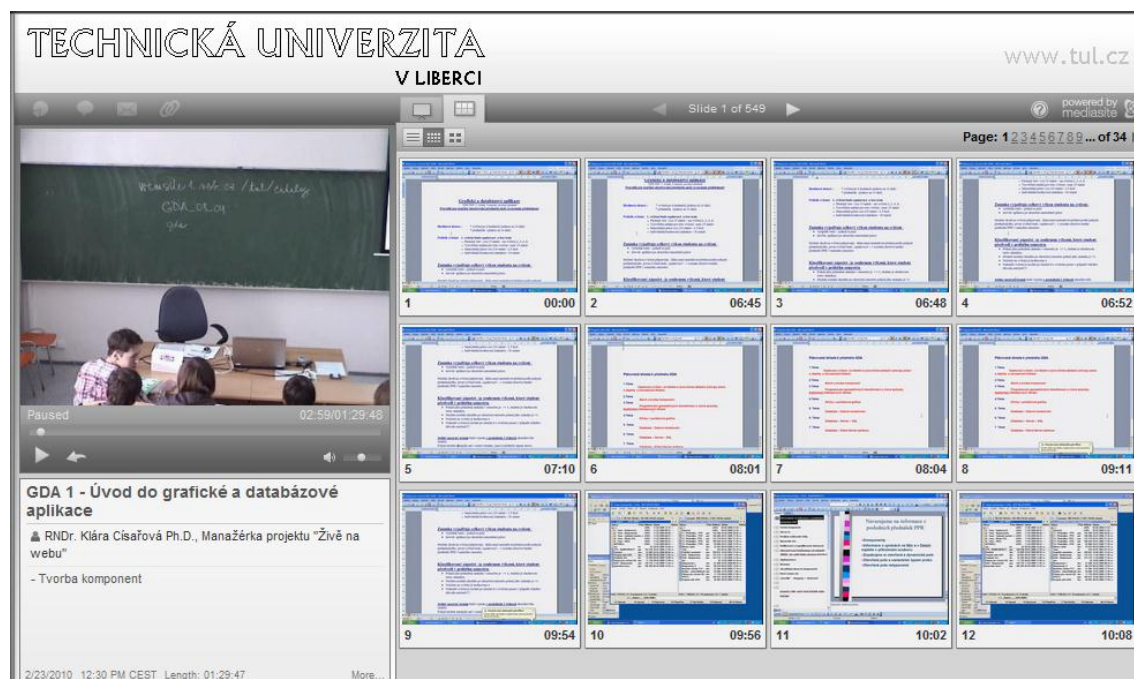


Obrázek 3: Stránka prezentace v Mediasite

Celým záznamem je možné se pohybovat posuvníkem ve spodní části okna videa, který zobrazuje časovou osu (viz Obrázek 3). Vpravo od videozáznamu může mít

uživatel buď zobrazen aktuální snímek, nebo se může přepnout do režimu procházení zmenšených náhledů. Každý snímek je časově indexován a díky tomu se poklepáním na příslušnou miniaturu může student přesunout na konkrétní část přednášky. Videozáznam i jednotlivé snímky lze libovolně zvětšovat nebo zmenšovat do celého okna.

Kromě procházení zmenšených náhledů prezentace je také možné procházet kapitolami celé prezentace, které odpovídají časovým indexům zmenšených náhledů. Díky tomu má student více možností, jak se s přednáškou seznámit a jak ji také pohodlnou formou zpracovat. Náhledové okno prezentace slouží k rychlé navigaci (viz Obrázek 4).



Obrázek 4: Náhledové okno prezentace

4 Moderní cesty porovnávání multimédií

Jedním z problémů organizace archivu je také vyhledávání podle multimediálních dat. Z těchto důvodů bylo nutné zabývat se také tímto problémem, i když ve vlastním řešení jej nakonec nebylo možno použít. Zapracování funkcionalit tohoto druhu lze v budoucnu očekávat v jiném kontextu a proto je v této práci zmínka o zjištěných systémech. Metody byly zkoumány také proto, že po rozkrytí šablony Mediasite prezentace, bylo jasné, že bude potřeba přeorganizovat sadu obrázků, která ve výsledku slouží pro navigaci v záznamu a pro zobrazení datového proudu. Problém podobnosti, či nepodobnosti obrazových dat byl proto zásadní. Také se uvažovalo o tom, jak hledat významné slidy (JPEG obrázky), které by v navigaci neměly chybět.

V dnešní době dochází k rozvoji nových metod v porovnávání a vyhledávání multimediálního obsahu. Největší nárok na aplikaci je možnost načíst velké množství souborů, které může jít až do statisíců. Skupina lidí kolem RNDr. Davida Nováka, PhD. na Masarykově univerzitě v Brně se zabývá tzv. obsahovým prohledáváním multimédií. Jedná se o metodu, která má slibné výsledky a nadějně vyhlídky do budoucna. Dnešní internetové vyhledávače pracují na principu zpracování textových informací, které jsou k multimédiím připojeny. Myšlenka projektu, kterým se tým z Brna zabývá je porovnávat multimediální obsah ať už na internetu, nebo kdekoli jinde, podle jeho obsahových informací. Dosavadní obsahové vyhledávače, které vznikaly, trpěly na tom, že neposkytovaly takovou kvalitu porovnávání, jakou by uživatel očekával, nebo nebyl systém schopen vyhledávat ve větším množství souborů.

Brněnští odborníci se nedávno podíleli na projektu ve spojení s firmou IBM, díky které dokázali zdokonalit své vyhledávací algoritmy a dotáhnout je do takové fáze, která by dovolovala komerční využití. Tím také vznikla databanka „Pixmac“, která má být první opravdu kvalitní aplikací pro obsahové vyhledávání a také má pootevřít dveře dalším kvalitním projektům 63[4].

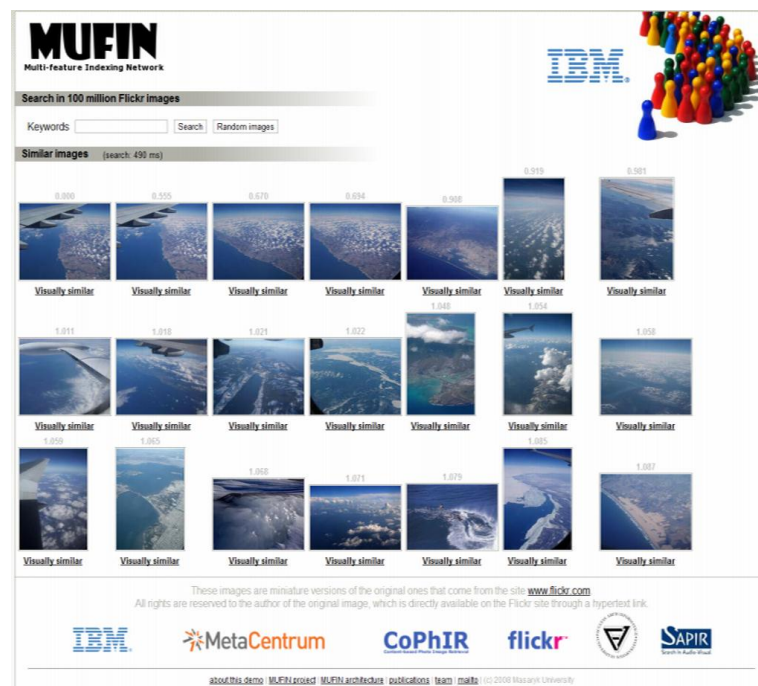
Oblastí využití, na které lze obsahové porovnávání multimédií aplikovat, je hned několik. Své uplatnění by takové aplikace našly v medicíně, kde by urychlily určení správné diagnózy. Další uplatnění je možné v e-shopech, kde by se porovnávaly podobné produkty.

4.1 Systém MUFIN

Dalším projektem, který tým z Brna připravuje ve spolupráci s firmou IBM a dalšími organizacemi, jako je například Grantová agentura České republiky, je webová aplikace MUFIN, která vyhledává podobnost v obrázcích na internetu a to podle tzv. „vzdálenosti“ obrázků. Zkratka MUFIN vyjadřuje MUlti-Feature Indexing Network, tedy síť indexování více vlastností. Základním principem podobnostního vyhledávání této aplikace je myšlenka, že objekty na dvou obrázcích jsou od sebe více či méně vzdáleny. Tato vzdálenost se počítá z vícedimenzionálního prostoru, do kterého se zanesou hodnoty, jako je například barevné spektrum, textura nebo rozložení tvarů v obrázcích. Čím blíže si potom body v tomto prostoru jsou, tím podobnější jsou si i testované obrázky. Funkce aplikace MUFIN se momentálně prakticky testují nad databází obrázků serveru Flickr, kde je systém schopný porovnávat více než 50 milionů obrázků mezi sebou. K výpočtům podobnosti a složení konečného výsledku se využívá peer-to-peer síť, která poskytuje možnost zpracovávání výsledků na více strojích paralelně. Díky tomu není problém porovnávat velké množství obrázků v řádech zlomků sekundy.

MUFIN pracuje s variabilní množinou tzv. distančních funkcí, které slouží k nalezení podobnosti. To je výhodné, protože lze říci, že momentálně neexistuje taková funkce, která by jednoznačně určovala podobnost obrázků. Tento způsob skrývá své kouzlo také pro uživatele, který si bude moci přesně definovat, která kritéria jsou pro něj při vyhledávání podobnosti důležitá. Může tak porovnávat například automobily podle tvaru, podle barvy apod. Celková podobnost je v tomto systému počítána jako vážená suma jednotlivých vlastností. Obrázky jsou předem zaindexovány, aby v budoucnu probíhal výpočet rychleji. Jejich index se přepočítává pouze v případě, že se například přidá nebo ubere některá z distančních funkcí.

Díky kvalitnímu způsobu výpočtu lze systém MUFIN taktéž bez obav využít i v medicíně, kde by si lékař mohl porovnat rentgenový snímek s velkým množstvím jiných a díky tomu odhalit nejružnější abnormality nebo komplikace u pacienta.



Obrázek 5: Rozhraní systému MUFIN

Rozhraní systému MUFIN lze připodobnit ke klasickému vyhledávači (viz Obrázek 5). Rozhraní je velice jednoduché a obsahuje pouze řádek pro zadání klíčového slova, podle kterého se má vyhledávat, pod ním se zobrazují výsledky porovnávání. Výsledky jsou řazeny podle toho, jakou mají vzdálenost od výchozího obrázku, který splňuje vyhledávané kritérium.

MUFIN řeší podobný problém, jako téma této práce. Tedy jak vhodným způsobem porovnat a určit míru podobnosti u velkého množství obrázků a tím vyloučit obrázky, které svojí podobností nemají pro prezentaci význam.

4.2 Metoda s využitím OCR

Velice moderní a zajímavou metodou řešení problémů s redundantními slidy by mohlo být využití OCR. Zkratka OCR vyjadřuje *Optical Character Recognition* a jedná se o přenos textu z tištěné podoby, případně obrázku opět do textové podoby. Metoda je založena na rozpoznávání znaků v obrázku podle jejich tvaru. Tato metoda není zcela stoprocentní například u obrázků, kde je text málo čitelný nebo pokud dojde ke spojení dvou písmen. V současné době metoda stále nedokáže stoprocentně převést obrázek na text, nicméně úspěšnost převodu se pohybuje mezi 70% - 95%. Tato úspěšnost by pro účely dodatečného vyhodnocení výsledných slidů v zamýšlené aplikaci byla adekvátní, ale jak bude dále uvedeno, není univerzální pro všechny druhy písma a nelze aplikovat

v případě, kdy dochází k deformaci písma na obrázku. K takové deformaci u webových prezentací často dochází. Díky tomu by se objevovaly chyby a úspěšnost metody by se radikálně snížila.

Snaha o rozpoznávání tištěného textu v podobném smyslu jako je OCR se poprvé objevila již kolem roku 1950, ale až v roce 1966 bylo standardizováno písmo OCR-A, které umožňovalo strojové čtení. Toto písmo bylo významným způsobem zjednodušené, aby stroje měly větší šanci znak rozpoznat a tím se zvětšila úspěšnost odečtu. Díky zjednodušení však došlo k tomu, že znaky byly hůře rozpoznatelné lidským okem. V roce 1968 se objevil další standard OCR-B, který již byl lépe čitelný lidským okem, ale nedosahoval tak úspěšných výsledků ve strojovém rozpoznávání.

Postup rozpoznávání v případě tištěného dokumentu lze rozdělit do dvou fází. Nejprve je nutné naskenovat text do počítače a vytvořit z něj obrázek. Po převodu na obrázek je počítač schopen rozeznat černé a bílé (popřípadě jinak barevné) body v předloze. V následující části dojde k analýze bloků, které k sobě logicky patří, protože dokument může být sestaven z většího množství tabulek, odstavců apod. Po rozdělení textu na jednotlivé části se může vykonat rozpoznávání znaků. Rozpoznávání lze provádět různými metodami, ale základní postupy jsou dva:

- rozpoznávání podle předlohové matice,
- rozpoznávání podle charakteristických tahů písmen.

4.2.1 Rozpoznávání podle předlohové matice

V tomto případě se využívá dané předlohy písma v digitalizované podobě a porovnává se s naskenovaným obrázkem. Pokud je tedy na obrázku písmo, které nemá rozpoznávač v paměti, nelze tento text převést. Jakákoliv deformace písma nebo jeho jiná degradace přináší do rozpoznávání zásadní chyby.

4.2.2 Rozpoznávání podle charakteristických tahů písmen

Při této metodě probíhá analýza jednotlivých částí písmen, a zjišťují se informace, zda jsou v písmenu přítomné oblouky, vodorovné nebo svislé čáry. Proces se nazývá topologická analýza. V tomto případě je metoda významně účinnější, protože pro analýzu není rozhodující velikost rastru ani například zúžení písmen.

V obou metodách je ovšem stále nutný zásah člověka, který musí opravit případné chyby vznikající v procesu rozpoznávání. Některé aplikace se dokážou „naučit“ nová písmena nebo překonat opakující se chyby v rozpoznávání, které mohou vzniknout. Pro řešení problémů s redundantními slidy v prezentacích není ale tato metoda natolik efektivní, jak bylo původně zamýšleno.

5 Šablona systému Mediasite

V této části bude detailně prezentována HTML šablona systému Mediasite. Budou představeny hlavní skripty, které jsou pro šablonu důležité, a také bude popsáno, jak přizpůsobit celý zdrojový kód řešení problému optimalizace navigace a redukce obrázků. Zdrojový kód je nutné upravit v tom smyslu, aby se v liště zmenšených náhledů zobrazovaly jen výsledné obrázky, které jsou pro obsah prezentace důležité, ale naproti tomu v plném zobrazení prezentace zůstaly veškeré slidy v původní nezměněné formě.

Hlavním spouštěcím souborem je *Player.html*, který definuje pouze vizuální prvky hlavní stránky systému Mediasite a obsahuje odkazy na dílčí skripty v jazyce *JavaScript*. Tyto skripty se starají o přehrávání videa a zobrazování snímků prezentace. Nejdůležitějšími, které se přímo váží k řešenému problému práce, jsou:

- *standalone-manifest.js* – zaštiťuje definici struktury slidů a jejich časování,
- *SlideBrowser.js* – zajišťuje zobrazení okna se zmenšenými náhledy obrázků,
- *MediaSitePlayer.js* – obstarává provázanost videa s jednotlivými snímky prezentace tak, aby se snímky střídaly podle konkrétního časového úseku videa,
- *Kernel.js* – definuje události v systému Mediasite,
- *MediaPlayer.js* – obsahuje metody, které rozšiřují systémové události v systému Mediasite a definuje chování nejrozličnějších ovládacích prvků.

Zbylé skripty, které zde nejsou zmíněny, fungují spíše jako podpůrné. Lze zmínit například skripty vztahující se k různým druhům přehrávačů podle typu operačního systému. Takové skripty tedy zajišťují přenositelnost videa na celou řadu platforem. Dalším souborem, který stojí za zmínku je skript obsahující metody zvětšení slidů na celou obrazovku.

5.1 Definice struktury slidů

Pro definici struktury slidů je v šabloně určen skript *standalone-manifest.js*. Stará se o uložení obecných informací o prezentaci jako je název, jméno přednášejícího, délka prezentace, datum a čas, kdy byl záznam pořízen. Dále se zde nachází odkazy na jednotlivé soubory jako je zdroj videa a obrázků, který se z prezentace zaznamenal.

Mimo jiné definuje, v jaké datové struktuře se budou ve zdrojovém textu slidy vyskytovat a jaké metody k nim budou náležet.

Kromě struktury snímků lze do skriptu zavést také kapitoly prezentace, které zpřístupní rychlejší uživatelský průchod prezentací.

Slidy jako takové jsou definovány ve formátu text, časová značka a popis, přičemž text a popis nejsou povinné. Vše je následně uloženo do proměnné *Slide*. Aby bylo docíleno návaznosti slidů, je z této proměnné vytvořeno pole slidů *Slides*, kde má každá položka odpovídající časovou značku. Proměnné *Slide* bude vhodně využito i při úpravě šablony.

Ve zmiňovaném skriptu se dále nachází metody, které se ke slidům vztahují:

- získání názvu a umístění slidu – *GetSlideURL*,
- získání časové značky slidu z pole – *GetSlideTime*,
- načtení obrázků nebo kapitol při spuštění a při zobrazení náhledů – *OnLoad*.

Všechny metody jsou hojně využívány v jiných skriptech například pro získání aktuálního obrázku prezentace při posunu videa nebo při výběru dané kapitoly.

5.2 Chování seznamu náhledů

Dalším důležitým skriptem je *SlideBrowser.js*. Obsahuje veškeré metody, které jsou svázány s náhledovým oknem aplikace, zobrazováním náhledových slidů, událostí *OnClick* na slidy apod. Skript dále obsahuje metodu pro stránkování zmenšených náhledů, a pro jejich návaznost. Dále metodu pro zobrazení aktuálního slidu, který odpovídá časové značce videa. V tomto případě se skript odkazuje do souboru *MediaPlayer.js* a *standalone-manifest.js* na již zmíněnou funkci *GetSlideURL*. Důležitou součástí je funkce, která umí přepínat mezi řádkovým výpisem náhledových slidů a stránkovým výpisem. S tím také souvisí sdružování obrázků do skupin po jednotlivých stránkách. Zajímavou metodou, kterou lze nalézt, je zoom při přejetí myši přes konkrétní slide, kdy se nejdříve zobrazí ikona lupy a poté se obrázek zvětší.

Stránkový seznam náhledů má dvě možnosti zobrazení, prvním je zobrazení menších obrázků v poli 4×3. Ve druhé variantě zobrazení se vyskytují větší obrázky v poli 3×2. Díky tomu má uživatel možnost získat větší perspektivu a orientaci v prezentaci a vybrat si to zobrazení, které mu vyhovuje. Další možností je zobrazení

kapitol prezentace. Za což je odpovědný také tento skript, protože jednotlivé odkazy na kapitoly spojuje s časovými značkami a jim příslušnými slidy.

5.3 Provázání videa a obrázků

Ve skriptu *MediaSitePlayer.js* dochází k provázání všech slidů s časovými úseky videa. Tento skript je také odpovědný za zobrazení multimediálních prvků na hlavní stránce systému Mediasite, tedy videa a zachyceného obrázku z prezentace (viz Obrázek 3). Ve skriptu se vyskytují i metody pro tlačítka přehrávače videa, kterými jsou zobrazení videa na celou obrazovku, pauza, obnovení videa nebo návrat na začátek prezentace. Zobrazený obrázek z prezentace lze mimo jiné zvětšit na celou obrazovku. Umístěny jsou zde definice pro vzhled jednotlivých tlačítek na hlavní stránce systému. Přímo z rozhraní lze také zaslat dotaz prezentujícímu nebo sdílet odkaz či sledovat statistiky prezentace. I za tuto funkčnost odpovídá tento soubor spolu s dalšími podpůrnými skripty.

Všechny tři předchozí skripty jsou ve větší míře propojené a řada funkcí je následně volána právě v souboru *MediaSitePlayer.js*, který je dále volán ve skriptu *Kernel.js*. Jedním z příkladů je vykreslování správného pořadí obrázků v okně pro snímek prezentace.

5.4 Definice systémových událostí Mediasite

V souboru *Kernel.js* jsou definovány události systému, které jsou v dalších skriptech rozšiřovány a využívány. Pomocí skriptu *Kernel.js* dochází například i k vhodné volbě přehrávače videa podle typu operačního systému, ale také do jakých stavů se může prezentace dostat, popřípadě co se stane ve chvíli, kdy prezentace skončí. Lze zde nalézt události, které se váží k navigaci v prezentaci, tedy definice metody, která se provede ve chvíli, kdy uživatel klikne například na tlačítko pro zobrazení následujícího slidu. Dalo by se říct, že právě tento skript je základním stavebním kamenem celého systému, který provazuje ostatní skripty mezi sebou.

5.5 Rozšíření systémových událostí

Ke spouštění videa a střídání slidů slouží skript *MediaPlayer.js*. Využívá metody dříve definované v *Kernel.js* a dále je svazuje s konkrétním chováním pomocí klíčového

slova *bind*. V uvedených metodách se například ověřuje, v jakém konkrétním stavu se prezentace nachází, jestli je ve stavu pauzy, nebo již ukončená, případně co se má provést ve chvíli, kdy nelze prezentaci načíst nebo je v jiném chybovém stavu. Dále lze nalézt metody pro ovládání hlasitosti videa nebo naprosté vypnutí zvuku ve videu. V konečné úpravě prezentace podle kritérií, které tato diplomová práce stanovuje, jsou nejdůležitější metody, které se starají o načtení správné části videa prezentace a také správného slidu při výběru kteréhokoliv ze zmenšených náhledů v náhledovém okně.

5.6 Úprava šablony pro redukci slidů

Úprava šablony probíhá v několika krocích, kdy se postupně upravují všechny z výše zmíněných skriptů. Kroky úpravy jsou následující:

- úprava struktury po redukci,
- úprava chování, zobrazení a akcí,
- úprava provázání prezentace,
- úprava systémových událostí a časování videa.

5.6.1 Úprava struktury po redukci

V prvním kroku se bude upravovat soubor *stanalone-manifest.js*. Do tohoto souboru je nutné přidat definici pole slidů, které vycházejí z redukce obrázků v celé prezentaci. Využije se již zmíněného datového typu *Slide* a vytvoří se nové pole *Slides_Thumbnail*, jehož velikost bude odpovídat počtu výsledných obrázků. Zároveň je nutné vytvořit do složky s obrázky v systému Mediasite novou složku, ve které budou umístěné výsledné slidy, složka je pojmenována jako *Thumbnails*. Do skriptu se přidá řádek, který bude odkazovat na nově vytvořenou složku.

Aby se při otevření náhledové části systému zobrazily správné slidy, je přidána ještě metoda *OnLoad*, která pracuje s novým polem a tedy s výslednými obrázky. Dojde také k přidání zmíněných funkcí *GetSlideUrl* a *GetSlideTime*, ale v tomto případě opět provázaných s odkazem na výsledné pole obrázků. Do nové struktury se ke každé položce přidá odpovídající časová značka. Nakonec se vytvoří ještě druhé pole pro kapitoly, které bude mít stejný rozměr i hodnoty jako pole *Slides_Thumbnails*. K této struktuře se vytvoří příslušná metoda *LoadChapters*, která seznam kapitol do systému načte.

5.6.2 Úprava chování, zobrazení a akcí

V souboru *SlideBrowser.js* je úprava pro zobrazení výsledného náhledu jednodušší. Zde je nutné vyhledat všechny výskyty odkazu na pole slidů z prvního upraveného skriptu a nahradit je odkazem na nově vytvořené pole výsledných obrázků. K nahrazení dojde pouze ve funkcích, které se vztahují k zobrazování slidů v náhledovém okně systému. Díky tomuto kroku se docílí požadovaného stavu, kdy se v hlavním okně systému budou zobrazovat všechny slidy a v náhledovém okně pouze ty výsledné.

V druhé části této úpravy se definuje odkaz na metodu *NavigateToSlide_Thumbnail*, která bude vytvořena v posledním kroku celkové úpravy v souboru *MediaPlayer.js*.

5.6.3 Úprava provázání prezentace

Aby se v náhledovém okně zobrazovaly správné obrázky výsledných slidů tak, jak jdou za sebou, je třeba upravit taktéž soubor *MediaSitePlayer.js*. Ve funkci, která obstarává zmíněné zobrazení náhledových slidů, je nutné změnit odkaz na nové pole s výslednými obrázky, aby se náhledy získávaly právě odsud a aby jim tedy odpovídaly i jejich časové značky.

Po těchto úpravách dojde k tomu, že se prezentace zobrazuje tak, aby měl uživatel možnost vybrat si mezi hlavními body prezentace, ale v hlavním okně může stále sledovat prezentaci v nezměněné formě.

5.6.4 Úprava systémových událostí a časování videa

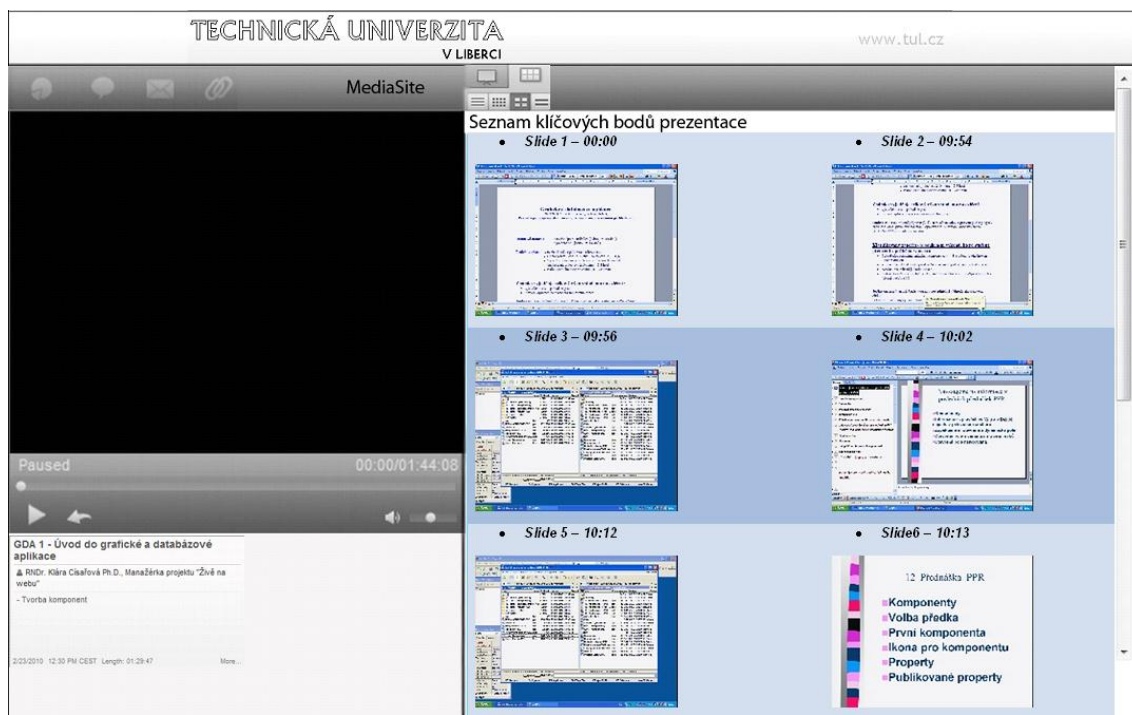
V posledním kroku úpravy se zajistí, aby se při kliknutí na nově vytvořené obrázky v náhledovém okně posunulo video na konkrétní čas a v neposlední řadě, aby se v okně aktuálního snímku prezentace zobrazil snímek, na který uživatel kliknul. Za tímto účelem se upravuje soubor *Kernel.js*, ve kterém se nadefinuje nová metoda *NavigateToSlide_Thumbnail*, jež bude strukturou odpovídat původní metodě *NavigateToSlide*. V druhé části této úpravy je nutné ve skriptu *MediaPlayer.js* definovat konkrétní chování metody *NavigateToSlide_Thumbnail*. I v tomto případě bude chování stejné jako původní metoda. V nové i původní metodě se využívá zadaných časových hodnot v poli slidů, aby se video posouvalo do správného časového intervalu.

Po těchto zmíněných úpravách jsou v náhledovém okně prezentace zobrazeny jen slidy předem vybrané aplikací RichMedia Slider. Tyto snímky se zobrazují pouze v optimálním počtu, který odpovídá klíčovým momentům prezentace. Další výhodou je celková přehlednost, protože například díky metodě *Zoom*, která je v systému definována si uživatel může zobrazit, co konkrétní náhledový snímek obsahuje a navigace v prezentaci je opět o něco jednodušší.

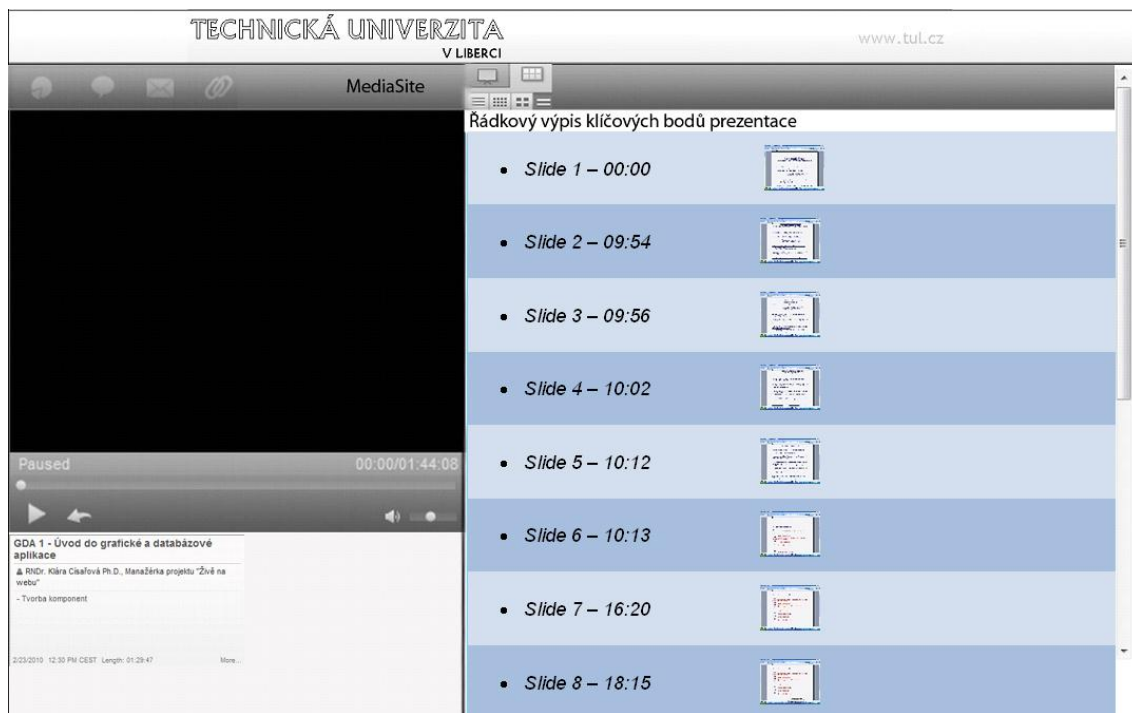
6 Návrh optimalizované navigace v Mediasite

V následující kapitole bude představen návrh navigace v systému Mediasite, který by přinesl optimalizaci ve webových prezentacích pro studenty. Návrh vychází ze současné verze systému Mediasite a je pouze částečně upraven. Z designového hlediska je zvětšena oblast pro zobrazování slidů z prezentace, ale zvětšení se dočkalo i okno videa prezentace. Inspirace pro zobrazení seznamu slidů a seznamu klíčových bodů v prezentaci vychází z designu a rozmístění navigace na serverech jako je „youtube.com“ nebo „stream.cz“. Video je umístěno vlevo, zatímco navigační část celého prostředí je situována v pravé části stránky. Do návrhu jsou přidány dvě možnosti zobrazení navigačního okna. V první řadě má student dostupný seznam všech slidů prezentace, kterými může volně listovat. V současné podobě systému Mediasite je zobrazení slidů rozvrženo do stránek. Z uživatelského hlediska by bylo výhodnější zobrazovat slidy v jakémsi *scrollboxu*, kterým by bylo možné se posouvat, a který by skutečně zobrazil všechny slidy bez nutnosti přepnutí stránky navigace. Zobrazení všech slidů přináší taktéž dvě verze náhledu a tím je řádkový výpis s malými náhledovými obrázky a dále maticové zobrazení $2 \times X$, které zobrazuje větší náhledy. Obě metody zobrazení jsou přístupné taktéž pro navigaci v již zmíněných klíčových bodech prezentace. Pro klíčové body prezentace by mohlo být vhodně využito výstupu, který nabízí aplikace vytvořená v rámci této diplomové práce.

Optimalizované zobrazení navigační lišty je důležité převážně z hlediska uživatelského komfortu studentů. Cílem je nabídnout takové rozhraní, ve kterém se rychle zorientují a získají rámcovou představu o obsahu přednášek. Jednoduše se poté mohou přepínat mezi jednotlivými částmi prezentací a tak získat ucelené znalosti předmětů, které studují. Návrhy zobrazení jsou jen rámcovým naznačením, jakým směrem by se mohl vývoj systému Mediasite ubírat. Optimalizované rozhraní bylo navrženo jak pro maticový (viz Obrázek 6), tak pro řádkový výpis (viz Obrázek 7). Všechny návrhy jsou přístupné v příloze této práce (viz Příloha B).



Obrázek 6: Maticový seznam klíčových bodů prezentace



Obrázek 7: Řádkový výpis klíčových bodů prezentací

7 Návrh aplikace RichMedia Slider

Návrh aplikace RichMedia Slider, která by měla splnit vytyčené cíle diplomové práce, zpřístupňuje metody porovnání libovolného počtu obrázků a zároveň vhodně pracuje s HTML šablonou systému Mediasite. Obrázky jsou vybírány ze zdrojové složky, do které jsou nasnímány v učebně. Pro porovnávání bitmap v programu RichMedia Slider jsou použity čtyři různé metody. První porovnává přímo pixely v bitmapě a závěrem vypíše, kolik rozdílných pixelů se v bitmapách vyskytuje, výsledek je uveden v procentech. Druhá metoda využívá výpočtu Euklidovské vzdálenosti v RGB prostoru krychle. Nakonec je tato hodnota, převedena na procenta a také vypsána. Další metodou je výpočet průměrného rozdílu hodnot jednotlivých pixelů, opět převedený na procenta. Poslední využitou metodou je kombinace, při které se předchozí metody smísí podle zadaných váhových koeficientů.

Pro porovnávání jsou nejdříve obrázky ve formátu JPEG převedeny na formát BMP, se kterými se dá jednoduše manipulovat a je možné přistupovat přímo k pixelům. Vzhledem k tomu, že je JPEG ztrátová komprese, promítne se tato ztráta i do výsledného zobrazení v BMP. Znaky ztrátové komprese však nesou všechny obrázky a výsledky jsou tudíž dostatečně objektivní.

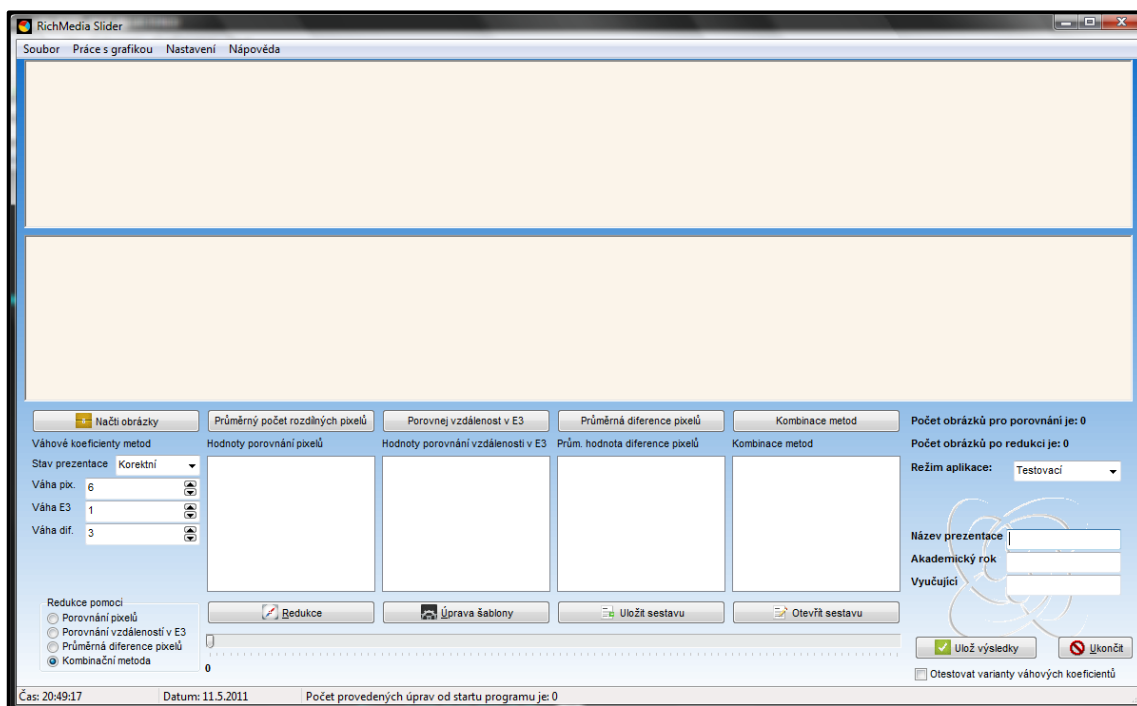
Další možností pro porovnání bitmap je například pomocí výpočtu kontrolních součtů (tzv. CRC) jednotlivých souborů. Tato metoda není pro toto zadání příliš vhodná, jak již bylo naznačeno v kapitole 4.2, protože nedosahuje natolik vysoké účinnosti a vyžadovala by si další uživatelský vstup. Porovnat obrázky lze také pomocí tzv. streamů, tato metoda dokáže určit, zda jsou obrázky totožné či nikoliv.

7.1 Uživatelské rozhraní

Uživatelské rozhraní je navrženo tak, aby mohl uživatel jednoduše a intuitivně testovat hodnoty porovnání bitmapových souborů a následně upravit HTML šablonu systému Mediasite. Načtené obrázky se zobrazí v posuvném *scrollboxu* a v dalším kroku může uživatel obrázky libovolně testovat a porovnávat.

Rozhraní je členěno do několika částí (viz Obrázek 8). V horní části se nachází zmíněný *scrollbox* pro zobrazení načtené sady obrázků. Druhý *scrollbox* se nachází níže a slouží pro zobrazení obrázků, které projdou redukcí. Pod *scrollboxem* výsledné sady

obrázků se nachází tlačítková lišta, která nabízí načtení obrázků a metody porovnání bitmap. Pod tlačítkovou lištou jsou zobrazeny čtyři *listboxy*, které po výpočtu porovnání zobrazí hodnoty rozdílů. Ve spodní části se dále nachází *trackbar*, kterým lze nastavit hodnotu pro redukování načtených bitmap. Nastavenou hodnotu musí uživatel potvrdit tlačítkem *Redukce*. Vlevo dole jsou umístěny čtyři *radiobuttony*, kterými lze vybrat, podle kterých výsledných hodnot porovnávání se má sada obrázků redukovat.



Obrázek 8: Uživatelé rozhraní aplikace

Dalšími důležitými ovládacími prvky jsou *spinedity*, kterými se nastavuje váhový koeficient příslušné metody porovnávání slidů do kombinační metody. Nad těmito *spinedity* lze nalézt *combobox*, kterým lze přepínat, zda se jedná o prezentaci zatíženou chybou či nikoliv. Ve většině případů provede přepnutí systém automaticky již při načtení snímků.

Kromě tlačítka „Redukce“ se pod zmíněnými výsledkovými *scrollboxy* nachází tlačítko „Úprava šablony“, kterým dojde k přepsání HTML šablony systému Mediasite pro využití nově redukované sady obrázků. Dále lze na stejném místě nalézt tlačítko pro uložení sestavy načtené sady obrázků a její následné otevření. Tyto funkce se hodí v případě, kdy uživatel potřebuje prostředí aplikace opustit a chce uložit rozpracovanou redukci.

V pravé části aplikace jsou umístěny shora informace o počtu načtených obrázků a dále informace o počtu obrázků po provedené redukci. Pod nimi je dále umístěn *combobox*, kterým lze vybírat režim aplikace. Na výběr je manuální režim, kdy uživatel musí

provést všechny kroky redukce ručně, dále automatický režim, kdy se pouze vybere obrázek, kterým prezentace začíná a aplikace sama propočítá jednotlivé metody rozdílnosti obrázků a provede redukci. Posledním režimem aplikace je testovací režim, při jehož výběru se zobrazí tlačítko „Ulož výsledky“, které ukládá spočítané rozdíly zobrazené v *listboxech* do textového souboru tak, aby bylo s těmito hodnotami možné pracovat a výsledky vhodně analyzovat. S tím souvisejí také textová pole, která jsou umístěna výše a do kterých se vyplňuje název prezentace, akademický rok, ve kterém byla prezentace vytvořena a také jméno vyučujícího, který předmět vedl. V případě, že uživatel zaškrtně *checkbox* „Otestovat varianty váhových koeficientů“, dojde k propočtu všech kombinací, kdy se liší váha metody průměrného počtu rozdílných pixelů a váha průměrného rozdílu pixelů. Váha vzdálenosti v prostoru RGB krychle byla vyhodnocena pouze jako doplňková hodnota, která do rozdílu výrazněji nepromlouvá. Tato pole jsou společně se zmíněným *checkboxem* a tlačítkem pro uložení výsledků do textového souboru dostupné pouze v testovacím režimu aplikace. Poslední tlačítko „Ukončit“ slouží k uzavření okna aplikace.

V horní části aplikace se nachází menu, kterým může uživatel aplikaci taktéž ovládat. V nabídce „Soubor“ jsou umístěna tlačítka pro načtení obrázků, uložení a otevření načtené sady a uzavření aplikace. Pod nabídkou „Práce s grafikou“ jsou umístěna tlačítka, která vyvolají výpočetní algoritmy pro porovnávání načtené sady obrázků.

Samostatnou částí je potom tlačítko v menu pro nastavení. Po jeho stisknutí se zobrazí nastavovací okno (viz Obrázek 9).



Obrázek 9: Nastavení formátu názvu obrázků

V levé části nastavovacího okna jsou umístěny dva *spinedity*, kterými se nastavuje počet znaků před a za číslem souboru. Pod *spinedity* se potom nachází text, který slouží jako příklad pro název souboru. Pod příkladovým textem jsou umístěny ještě další dva *spinedity*. První pro nastavení maximálního počtu optimalizovaných slidů v redukované

sestavě. Druhým lze určit zda se jedná o přednášku zatíženou chybou či nikoliv. Pokud přesáhne počet načtených slidů zadanou hodnotu, váhové koeficienty pro kombinační metodu výpočtu se změni na doporučené nastavení pro prezentace zatížené chybou. Po kliknutí na tlačítko „Ulož hodnotu“ se obě hodnoty uloží do proměnných a také do inicializačního souboru, aby byly přístupné i po dalším spuštění aplikace.

Poslední tlačítko v horním menu aplikace je tlačítko „O programu“, které zobrazí okno s informacemi o verzi a také s poučením o autorských právech, která se na aplikaci vztahují.

Na spodní liště aplikace může uživatel sledovat aktuální čas a datum, ale také, kolik úprav šablony systému Mediasite provedl od spuštění aplikace. Tato informace může posloužit pro statistické vyhodnocení, kolik úprav se provedlo například v daném měsíci.

7.2 Metoda průměrného počtu rozdílných pixelů

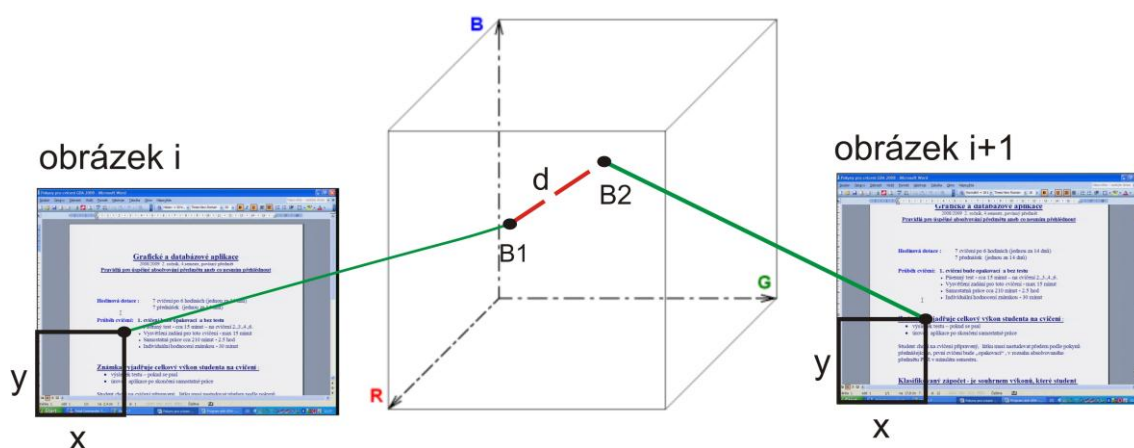
Pro tuto metodu je nutné, aby měly oba porovnávané obrázky stejné rozměry. V případě rozdílné velikosti je nutné oba obrázky velikostně sjednotit. Pro rychlejší výpočet je využito načítání pixelů po řádcích a tyto řádky se poté porovnávají. Algoritmus tak porovná tolik řádků, kolik má bitmapa pixelů na výšku. Porovnávání přímo po jednotlivých pixelech by bylo mnohem časově náročnější. V případě rozdílnosti pixelů se inkrementuje hodnota, která v konečném důsledku zobrazuje, kolik se v bitmapě vyskytuje rozdílných pixelů.

Výsledky jsou převedeny na procenta a poté jsou zobrazeny do *listboxu*, kterým je možné listovat a prohlížet hodnoty. Procentuální hodnoty je dosaženo pouhým vydělením celkového počtu rozdílných pixelů jejich celkovým počtem a vynásobeným stem. Po kliknutí na určitou položku *listboxu* se otevře zvětšený náhled konkrétního obrázku a uživatel může ověřit patřičnou rozdílnost.

7.3 Porovnání vzdálenosti v RGB prostoru

Další metoda pro porovnání zadaných obrázků využívá prostoru RGB. RGB prostor si lze zobrazit jako krychli, kde osu x představuje červená složka, osu y zelená složka a osu z složka modrá (viz Obrázek 10). Každá barva a její odstín je v tomto prostoru krychle reprezentována určitým bodem uvnitř prostoru. Algoritmus pracuje právě se vzdáleností mezi dvěma body uvnitř zmíněného prostoru krychle. V obou

obrázcích se pro daný barevný bod na stejném místě určeném souřadnicemi $[x,y]$ v E2 se z kódu barvy vypočítá poloha barevného bodu uvnitř prostoru RGB v $B1=E3$ ($r1,g1,b1$) a $B2=E3$ ($r2,g2,b2$). Spočítá se vzdálenost $d=(B1,B2)$ a tento postup se opakuje přes všechny pixely obou bitmap, vzdálenosti se nasčítají, aby se získala průměrná vzdálenost, mezi barvami všech pixelů. Průměrná hodnota je poté převedena na procentuální vyjádření a je zobrazena v patřičném *listboxu* pro další využití.



Obrázek 10: Vzdálenost v RGB prostoru krychle

7.4 Průměrná hodnota rozdílu pixelů

Při této metodě dojde k opětovnému načtení řádků pixelů v obrázku. Po načtení se provede rozdíl hodnoty kódu barvy odpovídajících pixelů obou bitmap. Výsledek se převede na absolutní hodnotu a připočte se do výsledné proměnné, která bude vyjadřovat součet všech rozdílů. V dalším kroku se výsledná proměnná vydělí počtem pixelů v bitmapě a získá se tak průměrná hodnota rozdílu pixelů. Nakonec jsou všechny výsledky porovnání vypsány do dalšího *listboxu* a je možné podle nich redukovat sadu obrázků.

7.5 Kombinační metoda výpočtu

Kombinační metoda je pojmenována podle způsobu výpočtu, který využívá všechny tři výše zmíněné metody porovnání načtených obrázků. V principu funguje tak, že podle nastavených váhových koeficientů spočítá ke každé dvojici obrázků vážený průměr všech tří metod. Každý váhový koeficient je možné nastavit v rozmezí 1 – 10, přičemž celkový součet koeficientů nesmí přesáhnout právě hodnotu 10. Díky tomu je možné pro různé typy přednášek upřednostnit tu metodu, která nejlépe vyhodnotí klíčové slidy prezentace a tuto hodnotu dále upřesnit o zbývající dvě metody porovnání.

V případě využití váhových koeficientů v poměru 6:1:3 by výpočet vypadal takto:

$$difference = ((6 * v1) + (1 * v2) + (3 * v3))/10 \quad (1)$$

$v1$ – výsledek metody průměrného počtu rozdílných pixelů

$v2$ – výsledek metody průměrné vzdálenosti uvnitř RGB krychle

$v3$ – výsledek metody průměrné hodnoty rozdílu pixelů

7.6 Redukce obrázků

Redukce slidů prezentace se provádí v několika krocích. Nejprve se načte sada obrázků z prezentace. Zde může uživatel vybrat všechny obrázky z celé složky nebo pro potřeby testování může zvolit jako počáteční obrázek jakýkoliv ve složce. Při načtení obrázků se tyto zobrazují v horním *scrollboxu* aplikace spolu s jejich pořadovým číslem a také tlačítkem, které zajistí promítnutí označeného obrázků do výsledků redukce.

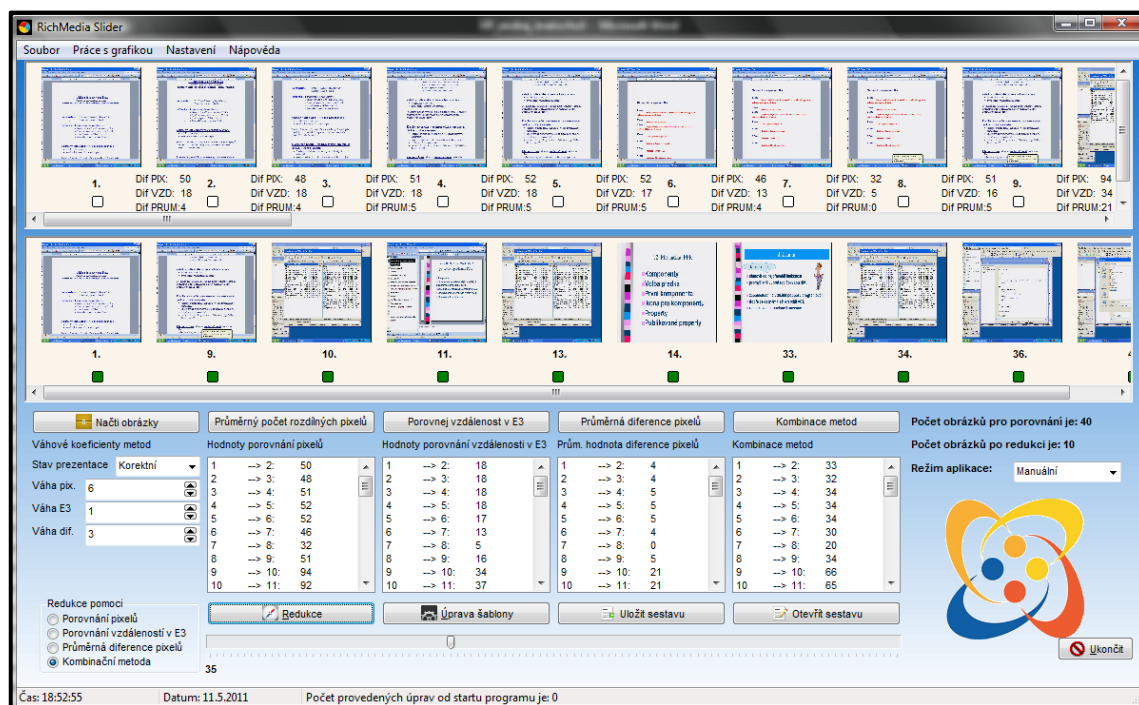
Následně má uživatel na výběr ze tří zmíněných metod porovnání nebo může stisknout tlačítko „Kombinace metod“, kdy se postupně propočítají všechny tři základní metody a podle nastavených váhových koeficientů se spočítá jejich výsledek. Kliknutím na zmenšený obrázek nebo na řádek uvnitř výsledkového *listboxu* uživatel aktivuje okno zvětšeného náhledu, ve kterém se lze pohybovat a prohlížet vybranou sadu obrázků.

V následujícím kroku je nutné nastavit hodnotu redukčního koeficientu na *trackbaru* zcela dole. Hodnota, která je přednastavena vyjadřuje nejčastější výskyt výsledků zvětšený o jednotku. Pomocí *radiobuttonů* vlevo může uživatel přepnout na jinou metodu porovnání, nad kterou by se měla vstupní skupina obrázků redukovat. Nyní je možné vybrat tlačítko „Redukce“, po jehož stisknutí se ve spodním *scrollboxu* zobrazí výsledná sada snímků. Každý obrázek má opět pořadové číslo a také tlačítko, které v tomto případě slouží k vynechání obrázku v upravené šabloně systému Mediasite.

V posledním kroku se kliknutím na tlačítko „Úprava šablony“ provede úprava HTML šablony systému Mediasite a jejích „javascriptových“ souborů přesně tak, jak je napsáno v kapitole 5.6.

Po úspěšné redukci v programu RichMedia Slider je možné ihned spustit prezentaci Mediasite, ve které se již zobrazí optimalizovaná sada obrázků v navigačním

okně. Náhled aplikace RichMedia Slider s načtenými vstupními obrázky a výslednou sadou je zobrazen na obrázku 11.



Obrázek 11: Provedená redukce

7.7 Použité metody a techniky

V průběhu vývoje aplikace vzešlo několik zajímavých úskalí a problémů, které bylo nutné vyřešit. V následující kapitole jsou rozebrány zajímavé části zdrojového kódu aplikace a také způsoby, kterými se povedlo problémy odstranit. Ukázky jsou doplněny o úryvky zdrojového kódu pro větší názornost. Zdrojový kód je dále k nahlédnutí na příloženém CD.

7.7.1 Gradient na vykreslení formuláře

V rámci uživatelského pohodlí a také vylepšení vzhledu aplikace byla zvolena komponenta *XpStyle*, která je přímo přístupná v prostředí Delphi 7. Tato komponenta pracuje s vizuálními vlastnostmi prvků a lze tak docílit kvalitnějších grafických vlastností uživatelského rozhraní. Největším rozdílem použití zmíněné komponenty oproti normálnímu vzhledu formuláře je vzhled tlačítek, která nemají tak ostrý tvar a plně využívají grafického potenciálu operačního systému.

Dalším krokem k navržení moderního uživatelského rozhraní bylo zvolení gradientní barvy na pozadí formuláře. V podstatě se jedná o to, aby na horní straně

pozadí byla barva modrá, která od horní části postupně světlá až na spodní straně je barva pouze bílá. Plynulý přechod zajišťuje volitelný interval kroků, ve kterém se jednotlivé barevné úrovně v řádcích pozadí propojí. Je využito procedury *FormPaint*, která se stará o vykreslování formuláře aplikace.

V proceduře je nejprve zvolen interval, který je zde nastaven na hodnotu 100. Nastavená hodnota postačuje i pro větší rozlišení monitorů. Dále je nutné nastavit jednotlivé RGB složky barvy, která se bude na horní části okna zobrazovat. Poté dojde v každém ze 100 kroků k připočtení přírůstku ke každé složce, aby bylo dosaženo efektu plynulého přechodu. Tímto způsobem vzniknou pásy s různými odstíny modré barvy, které se do formuláře vykreslí. Algoritmus pro vykreslení gradientu vypadá takto:

```
procedure TFm_Main.FormPaint(Sender: TObject);  
const btN = 100; { Number of steps }  
var  
    iI: Integer;  
    Col: TColor;  
begin  
    for iI := 0 to btN - 1 do  
        with Canvas do begin  
            Col := RGB(Round(24 + 231 * (iI / btN)), Round(116 + 139 *  
(iI / btN)), Round(205 + 50 * (iI / btN)));  
            Pen.Color := Col;  
            Brush.Color := Col;  
            Rectangle(0, Round(ClientHeight * (iI / btN)),  
ClientWidth, Round(ClientHeight * ((iI + 1) / btN)));  
        end;  
    end;
```

7.7.2 Zobrazení obrázku z náhledové lišty

Po vygenerování náhledové lišty všech načtených obrázků do *scrollboxu* bylo nutné vytvořit po kliknutí na libovolný obrázek proceduru, která daný obrázek zobrazí v plné velikosti. Pokud na zobrazený obrázek uživatel znovu klikne, přepne se na další v sadě. Přepínání bude pokračovat dokud se nedosáhne konce složky. Důležité je také

zobrazit zvětšený obrázek tak, aby byl v aktivním okně a aby hlavní okno aplikace bylo ve stejné chvíli na pozadí.

Veškeré obrázky ze zmíněného *scrollboxu* jsou generovány dynamicky, protože při každém načtení nové sady pracuje uživatel s různým počtem slidů. Pro hromadnou práci se všemi obrázky ve *scrollboxu* je vytvořeno dynamické pole, do kterého se obrázky načtou. Každému obrázku je přiřazena odezva na událost *OnClick* tak, jak je naznačeno v následující ukázce kódu:

```
(Dx = 105; Me = 16; TopOb = 5;)  
With pole[index] do begin  
  
    Parent:=Sb_nahled;  
  
    top:=TopOb;  
  
    left:=(index*Dx+(index*Me))+Me;  
  
    Width:=Dx;  
  
    Height:=Dx;  
  
    Picture.Assign(Jpg);  
  
    Stretch:=true;  
  
    Hint:=IntToStr(index+1);  
  
    OnClick:=Im_Click;  
  
end;
```

Po vykreslení zvětšeného obrázku se lze poklikáním do libovolného místa okna posunout na další, což zajišťuje následující kód:

```
procedure TFm_main.Nahledy(Sender:TObject);  
var cislo : Integer;  
begin  
    cislo:=StrToInt(Fm_nahled.Caption);  
    inc(cislo);  
    Nactisoubory(GNazev,cislo);  
if not FileExists(GVystup) then ShowMessage('Dosáhli jste konce složky!')  
else begin  
    nahled.Picture.LoadFromFile(GVystup);  
    Fm_nahled.Caption:=IntToStr(cislo);  
end;
```

`end;`

`end;`

Z předchozí ukázky je patrné, že k identifikaci pozice v rámci pole a přechodu na další položku je využito nadpisu okna zvětšeného náhledu, kde je tato hodnota přístupná v textovém formátu. K předávání hodnoty například při otevírání náhledového okna je využito vlastnosti *Hint* jednotlivých obrázků, do které je pozice v rámci pole obrázků uložena.

7.7.3 Načítání sady slidů

Před vlastním načtením sady slidů se nejprve provede kontrola uživatelem vybrané složky. Dalším krokem se pomocí funkce *pocetobrazku* získá hodnota celkového počtu slidů ve složce. Podle hodnoty počtu obrázků se nastaví dynamická pole, která se postarají jak o sadu zobrazených slidů ve *scrollboxu*, tak o popisky obrázků a další tlačítka pro přímé vybrání slidů do redukce. Poté se v cyklu provede načtení obrázků podle jejich pořadí ve složce, kde je nutné nejdříve ke každému obrázku dodat formát jeho očíslování. Formát je ve tvaru čtyř znaků, které číslo definují, a získává se následujícím algoritmem:

```
id:=cyklus+index+1;
StrId:= IntToStr(Id);
n:=Length(StrId);n:=4-n; cast:='';
    for i:=1 to n do cast:=cast+'0';
cast:= cast + StrId;
```

Po získání názvu příslušného slidu dojde k ověření, zda skutečně takový obrázek ve složce existuje. Nakonec se provede vykreslení obrázků, jejich popisků a tlačítek pro vybrání do *scrollboxu* načtené sady, kde jsou jednotlivé obrázky přehledně zobrazeny.

7.7.4 Zobrazení checkboxů v náhledové liště

S načítáním obrázků úzce souvisí i zobrazování *checkboxů* pro výběr takových slidů, které si uživatel přeje zobrazit ve výsledné sadě. V předchozí verzi aplikace docházelo k tomu, že se dynamické pole *checkboxů* přestalo vykreslovat, protože došlo k dosažení jejich maximální hodnoty vlastnosti *left*. Z toho důvodu byly *checkboxy*

nahrazeny objekty typu *shape* ve tvaru obdélníků, které reagují na událost *OnClick* tak, že změni svou barvu a jejich hodnota *Tag* změni hodnotu. V případě, že nejsou zatrženy, má jejich vlastnost *Tag* hodnotu 0 nebo hodnotu 1 v případě jejich zatržení. Podobnou vlastnosti jako *checkbox* vykazují taktéž komponenty *radiobutton*.

7.7.5 Funkce pro zjištění počtu obrázků

Funkce pro zjištění počtu obrázků v uživatelem vybrané složce je důležitá v celém zdrojovém kódu aplikace. Výsledek této funkce je uložen do globální proměnné *Gpocet* a tato proměnná vystupuje ve velké části využitých algoritmů. Funkce pracuje tak, že otevře vybranou složku a prochází veškeré obsažené soubory, které nejprve kontroluje, zda obsahují správnou příponu. Kontrola se provádí proto, že složky slidů v systému Mediasite často obsahují také soubor s videem nebo další doplňkové soubory, díky kterým by počet obrázků nebyl korektní. V dalším kroku zjištění počtu obrázků se provede ještě kontrola jejich názvu. V případě, že by se ve složce z jakéhokoliv důvodu vyskytoval obrázek, který k dané prezentaci nepatří, počet obrázků by byl opět nekorektní. Kontrola názvů obrázků se provádí podle kritérií, které je možné ovlivnit v nastavovacím okně aplikace (viz Obrázek 9). V případě splnění všech kritérií dojde k inkrementaci počítadla a jeho konečná hodnota je funkcí vrácena. Algoritmus vypadá následovně:

```

if ((hledani.Attr and faDirectory) <> faDirectory) then begin
    pripona:=copy(hledani.Name,length(hledani.Name)-2,4);
    if pripona = 'jpg' then
        begin
            try
                pos:=(StrToInt(copy(hledani.Name,length(hledani.Name)-
                    Gcislovani,4)))-1;
                if pos >= GPocatek then inc(pocitadlo);
            except on E:Exception do begin
                MessageBox(0,Pchar('Nekorektní          formát          názvu          souboru
                    '+hledani.Name+'!'),'RichMedia Slider',MB_ICONWARNING);
            end;
        end;
    end;

```



```

end;
until FindNext(hledani) <> 0;
FindClose(hledani);

```

7.7.6 Algoritmy porovnávacích metod

Algoritmy porovnávacích metod jsou hned po načítání velké sady obrázků nejnáročnějšími výpočty na paměť a vytížení počítače. Všechny porovnávací algoritmy mají velice podobný základ a liší se především v jádrech výpočtů, které vychází z teorie popsané v kapitolách 7.2, 7.3, 7.4 a 7.5. První tři algoritmy začínají načtením obrázku a jeho převedením na bitmapu pomocí procedury *prevod*. Tato procedura je dále popsána v literatuře [12]. Vždy se načítají první dva obrázky z načtené sady, které se po řádcích nasnímají. První algoritmus vypadá takto:

```

for j:=0 to YY-1 do
begin
    sl1 := BMP1.ScanLine[j];
    sl2 := BMP2.ScanLine[j];
    for i:=0 to XX-1 do
    begin
        if sl1[i]<>sl2[i] then inc(diference);
    end;
end;

```

Ve druhé výpočetní metodě se porovnává vzdálenost uvnitř RGB krychle. Porovnávání probíhá pomocí následujícího kódu:

```

for y := 0 to YY-1 do begin
    P := BMP1.Scanline[y];
    Q := BMP2.Scanline[y];
    for x := 0 to XX-1 do begin
        Diff := Diff + Sqr(P^ - Q^); inc(P); inc(Q);
        Diff := Diff + Sqr(P^ - Q^); inc(P); inc(Q);
        Diff := Diff + Sqr(P^ - Q^); inc(P); inc(Q);
    end;
end;

```

```
vysl := Sqrt(Diff / (XX*YY));
vysl := (vysl/(255*sqrt(3)))*100;
```

Ve třetí výpočetní metodě se počítá přímo s hodnotami pixelů. Provede se rozdíl pixelů na stejné pozici ve dvou po sobě jdoucích obrázcích tak, jak naznačuje následující algoritmus:

```
for j:=0 to YY-1 do
  begin
    sl1 := BMP1.ScanLine[j];
    sl2 := BMP2.ScanLine[j];
    for i:=0 to XX-1 do
      begin
        if sl1[i]<>sl2[i] then begin
          difference:=(difference + ABS(sl1[i]-sl2[i]));
        end;
      end;
    end;
```

V poslední kombinační metodě dojde ke spojení výsledků podle váhových koeficientů, které jsou nastaveny uživatelem. Výpočet, který se provádí v cyklu podle počtu načtených obrázků, lze ve zdrojovém kódu nalézt takto:

```
difference:=round(((vysledek_pix[index]*SE_pixely.Value)+(vysledek_k_E3[index]*SE_E3.Value)+(vysledek_prumer[index]*SE_Dif.Value))/10);
```

7.7.7 Redukce sady slidů

V algoritmu pro redukci sady načtených slidů se v cyklu provede kontrola na hodnotu nastavenou na *trackbaru* umístěném v aplikaci. Tuto hodnotu nastavuje uživatel, ale je mu doporučena hodnota podle nejčastější hodnoty zvolené metody. U každého slidu dojde poté k porovnání, zda tuto hodnotu splňuje nebo převyšuje, přičemž první a poslední slide se projeví do jakékoliv redukce. Dále se kontroluje, zda si uživatel některý z obrázků již předem nevybral a takový obrázek je nutné také ve výsledné sadě zobrazit. Kontrola pro redukci se provádí podle následující podmínky:

```
if (vysledky[index] >= uroven) or (index=(Gpocet-1)) or
(index=0) or (polecheck[index].Tag=1) then begin
```

Po splnění podmínky dojde k nastavení velikosti dynamických polí pro zobrazení obrázků, jejich označení a také se opět zobrazí tlačítka pro uživatelský výběr jednotlivých obrázků. Tato tlačítka tentokrát slouží uživateli k vyřazení nebo konečné korekci sady, která se zapíše do šablony systému Mediasite.

Po vygenerování výsledné sady snímků je následně provedena kontrola počtu výsledných obrázků podle hodnoty, kterou lze opět nastavit v nastavovacím okně aplikace (viz Obrázek 9). Uživatel je upozorněn nabídkou, zda si přeje upravit výsledek redukce na takový počet výsledných slidů, který je menší nebo roven zadané hodnotě. Pokud uživatel okno potvrdí, provede se opětovná redukce a nalezne se taková hodnota úrovně redukce na *trackbaru*, která vygeneruje kýžený počet obrázků. I v tomto případě je brán ohled na obrázky předem vybrané uživatelem.

7.7.8 Překreslení okna při náročných operacích

Při náročnějších výpočtech, které trvají delší dobu, jako je například načtení velkého množství obrázků, které jde řádově do tisíce položek nebo při jejich porovnávání se vyskytl problém, že aplikace přestala odpovídat. V takovém případě je možný dvojitý přístup. V první řadě lze vytvořit další vlákno aplikace, které se o výpočet samo postará. V tomto případě bylo ovšem využito procedury *Application.ProcessMessages*, která v malé, ale tolerovatelné míře zpomalí výpočetní mechanismy, nicméně dovolí překreslování hlavního aplikačního okna. Díky tomu jsou v kterémkoliv okamžiku dostupné veškeré prvky na hlavním formuláři a nedojde tak k situaci, že by aplikace přestala odpovídat. S využitím této procedury ovšem vznikl střet v tom smyslu, že uživatel mohl v průběhu výpočtu kliknout například na tlačítko jiného výpočtu a ten se začal paralelně také provádět. Z tohoto důvodu bylo nutné nastavit při výpočtech vlastnost *Enabled* hlavního formuláře na hodnotu *false*. Po výpočtu je tato hodnota opět vrácena na hodnotu *true* a uživatel může využívat všechny ovládací prvky aplikace.

Procedura *Application.ProcessMessages* zpracovává všechny čekající systémové zprávy aplikace a stará se například o posouvání okna a jeho celkové vykreslování ve chvíli výpočtu. Takové chování je přívětivější i z hlediska uživatele, který má jistotu, že aplikace stále pracuje a nedošlo k neočekávanému ukončení nebo pádu.

7.7.9 Šetrná práce s pamětí při výpočtech

Hlavním požadavkem na aplikaci RichMedia Slider byla již od počátku možnost načítat velké množství obrázků. Některé prezentace, které jsou dosud již nasnímané, obsahují i tisíc a více slidů a přesně tyto prezentace je dobré v rámci post-processingu v aplikaci zpracovat. Načítání takto velkého počtu obrázku s sebou nese jak náročnost pro procesor, tak i náročnost na paměťové prostředky počítače. V prvních verzích aplikace bylo použito modelu, kdy se všechny obrázky načítaly do paměti a díky tomu byly poté porovnávací výpočty velice rychlé. To se ale postupem času ukázalo jako špatná cesta, protože po načtení stovek obrázků systém proces ukončil se zprávou, že operační paměť je plná. Bylo tedy nutné se vydat takovou cestou, která by zachovala rychlost načítání slidů, ale zároveň udržela rychlost výpočtu na přijatelné míře.

Slidy se v aplikaci nyní načítají sekvenčně v cyklu, jak bylo naznačeno v kapitole 7.7.3. V každém cyklu se obrázek načte do proměnné *Jpg*. Poté se provedou činnosti popsané ve stejné kapitole a nakonec se proměnná *Jpg* uvolní. Díky tomu není paměť zbytečně zatížena obrázky v plné velikosti.

Spolu s tímto postupem bylo zamýšleno využít zmenšené obrázky pro výpočty hodnot porovnávacích algoritmů. To se ale ukázalo nevyužitelné, protože zmenšené obrázky nedokážou poskytnout natolik objektivní výsledky při porovnávání. Obrázky se tedy při porovnávání otevírají, porovnají a znovu uvolní, přičemž je kladen důraz na optimalizaci kódu, aby se některé obrázky neotevíraly vícekrát apod.

Doplňkové formuláře a dialogy se také po uzavření uvolňují z paměti, aby nezabíraly cenné paměťové místo.

7.8 Test na konkrétních typech přednášek

V této kapitole budou podrobně popsány testy na různých typech přednášek, podle kterých byly poté zvoleny některé vnitřní doporučené hodnoty aplikace RichMedia Slider. Počet slidů v konkrétních přednáškách je závislý na využití zařízení v učebně poskytující streamování přednášek, ale také na kalibraci snímacího mechanismu. Kritickým momentem může být špatně nastavený projektor, který způsobuje velkou redundanci slidů v prezentaci. I tento problém by měla navržená aplikace automaticky odbourat. Na základě testů byly vybrány hodnoty pro nastavení

počtu slidů, který určí hranici, zda je prezentace v pořádku a zda nedošlo například ke špatnému nasnímání vinou projektoru. Podle informace, zda je prezentace v pořádku, či je zatížena podobnou chybou, dojde k automatickému nastavení váhových koeficientů pro kombinační metodu porovnání, které vycházejí z testování prezentací. Všechny hodnoty, které vstupují do testů, byly vygenerovány pomocí testovacího režimu aplikace RichMedia Slider.

Celkem byl test proveden na deseti prezentacích. Výsledky testů jsou dostupné na přiloženém CD. Hlavními druhy prezentací, na kterých se porovnávací metody testovaly, jsou tyto:

- PP – přednes a prezentace,
- PIT – přednášky IT předmětů doplněné ukázkami práce na počítači, programování, prostředí case systémů, databázových serverů apod.,
- PVP – přednášky a výklad pojmů za využití snímacího zařízení, které snímá výklad na speciální tabuli nebo výklad za pomoci vizualizéru.

Typy přednášek jsou značeny podle [3].

Na grafech, které se nacházejí na následujících stránkách, jsou zobrazeny křivky popisující jednotlivé rozdíly slidů v průběhu prezentace spolu s konečnými výslednými slidy. Jednotlivé grafy jsou podle metod porovnání zobrazeny v tomto pořadí:

1. Průměrný počet rozdílných pixelů
2. Průměrný rozdíl pixelů
3. Kombinační metoda s váhovými koeficienty

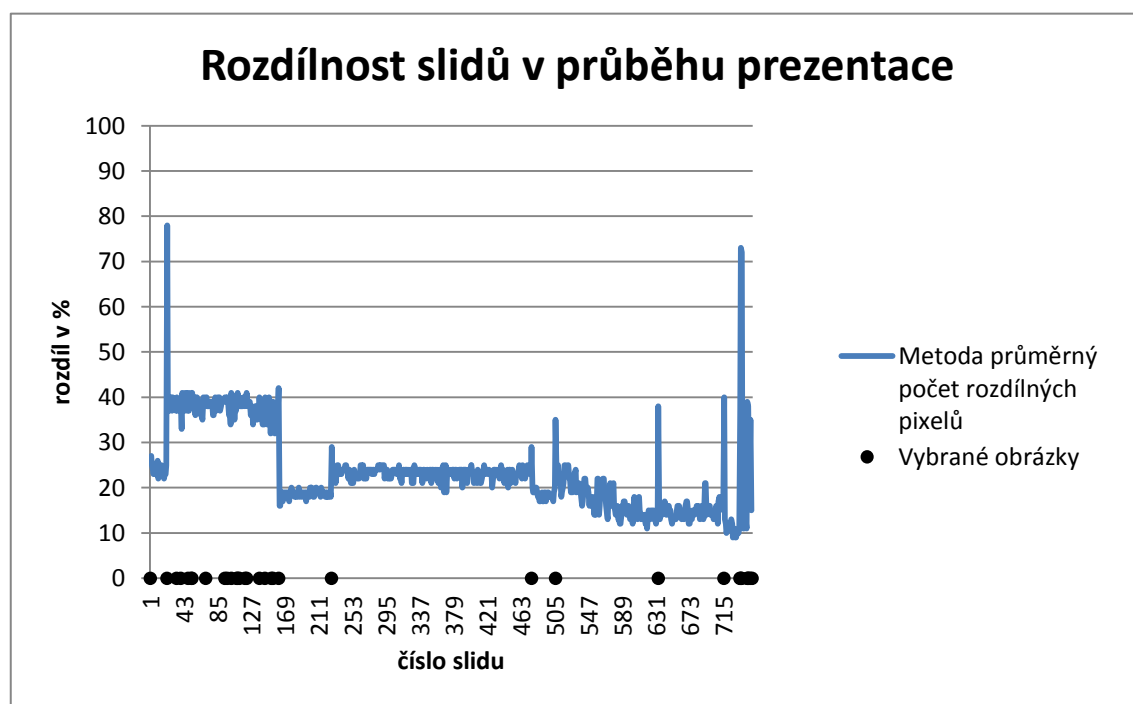
Na základě výstupů se ukázalo, že metoda porovnání vzdáleností uvnitř RGB krychle nedosahuje tak výrazných výsledků jako ostatní dvě metody. V důsledku tohoto pozorování byla této metodě přiřazena váha 1 pro všechny typy prezentací. Tato metoda výrazněji do rozdílnosti slidů nepromlouvá a její hodnoty a grafy k jednotlivým prezentacím jsou opět přístupné na přiloženém CD.

V testovacím režimu aplikace byly dále vygenerovány výsledky všech kombinací váhových koeficientů zbylých dvou metod. Pro prezentace zatížené chybou zařízení a významnou redundancí slidů byly určeny doporučené váhové koeficienty 2:1:7 v pořadí metod zapsaných výše. Pro prezentace, které nejsou chybou zatíženy, jsou určeny doporučené váhové koeficienty 6:1:3 v tomtéž pořadí. Tyto hodnoty jsou pouze doporučené a uživatel je může v aplikaci kdykoliv měnit.

Na základě porovnání snímků je výsledkem sada obrázků, která je do grafu taktéž zanesena. Podstatou je, aby se slidy, které tvoří klíčové momenty prezentace, projeví ve výsledné sadě. Podle nastavení hraniční hodnoty počtu výsledných slidů se klíčové obrázky vhodně doplní zbylými tak, aby nepřesáhly nastavenou hranici. Pro testování byla hraniční hodnota počtu slidů nastavena na 40. Hodnota počtu slidů, která vyjadřuje, zda je prezentace zatížena chybou, byla nastavena na 150.

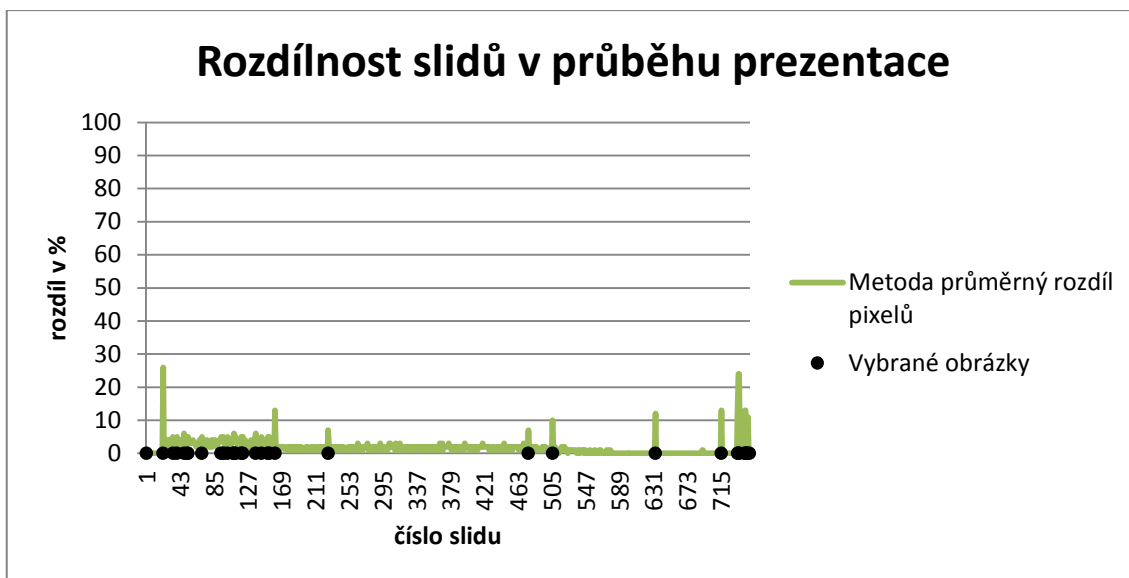
7.8.1 Test PP – přednes a prezentace

Pro test tohoto typu prezentace byla zvolena přednáška předmětu Algoritmizace a programování z akademického roku 2009/2010. Přednáška obsahuje 751 slidů. Tato prezentace byla zajímavá v tom, že byla zároveň zatížena již zmíněnou chybou snímání obrazu zobrazeného na projektoru a díky tomu vzniklo v prezentaci několik desítek redundantních slidů.



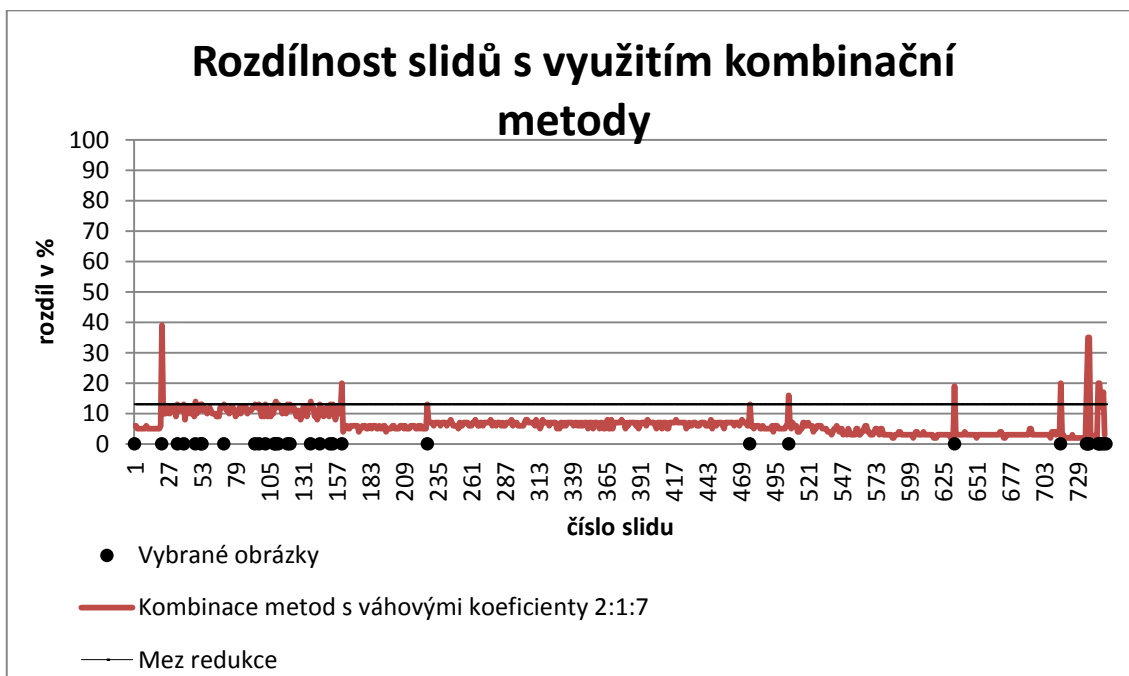
Graf 1: Rozdílnost slidů prezentace ALG 1. metodou

Z předchozího grafu je patrné, že touto metodou lze určit dva hlavní momenty prezentace, které lze vybrat mezi vybrané slidy. Ostatní obrázky jsou si svou rozdílností velmi podobné a to díky tomu, že se jedná o stejný slide, který se nasnímal několikrát za sebou. Chyba projektoru se projevuje „třesem“ obrazu, který na nasnímaných slidech způsobuje jejich rozdílnost.



Graf 2: Rozdílnost slidů prezentace ALG 3. Metodou

Na grafu 2 lze nalézt více bodů, které určují klíčové momenty prezentací. Podrobnějším zkoumáním slidů prezentace bylo ověřeno, že se skutečně jedná o slidy s rozdílným obsahem. Tyto rozdílné obrázky je potřebné taktéž promítnout do konečné sady vybraných slidů. Z předchozího tvrzení vyplývá, že na tuto metodu musí být kladena nejvyšší váha, protože dokáže s vyšší úspěšností rozlišit klíčové momenty prezentace.



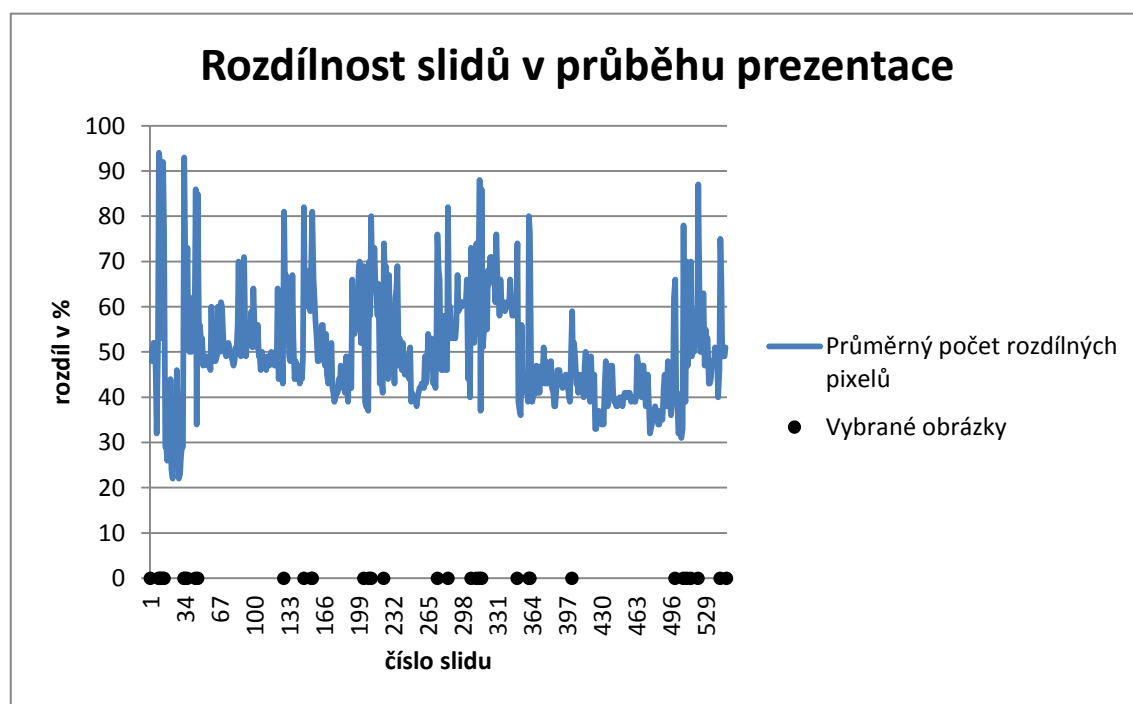
Graf 3: Rozdílnost slidů prezentace ALG kombinační metodou

Graf 3 zobrazuje rozdílnost pixelů za pomoci kombinační metody v poměru váhových koeficientů 2:1:7. Z grafu lze vyčíst, že do konečné vybrané sady byly

vybrány slidy, které jsou obsahově nebo graficky jiné a tedy v tomto případě dokáže aplikace automaticky překonat také prezentace zatížené chybou nebo přílišnou redundancí obrázků. Z původního počtu 751 slidů bylo vybráno 34 obrázků.

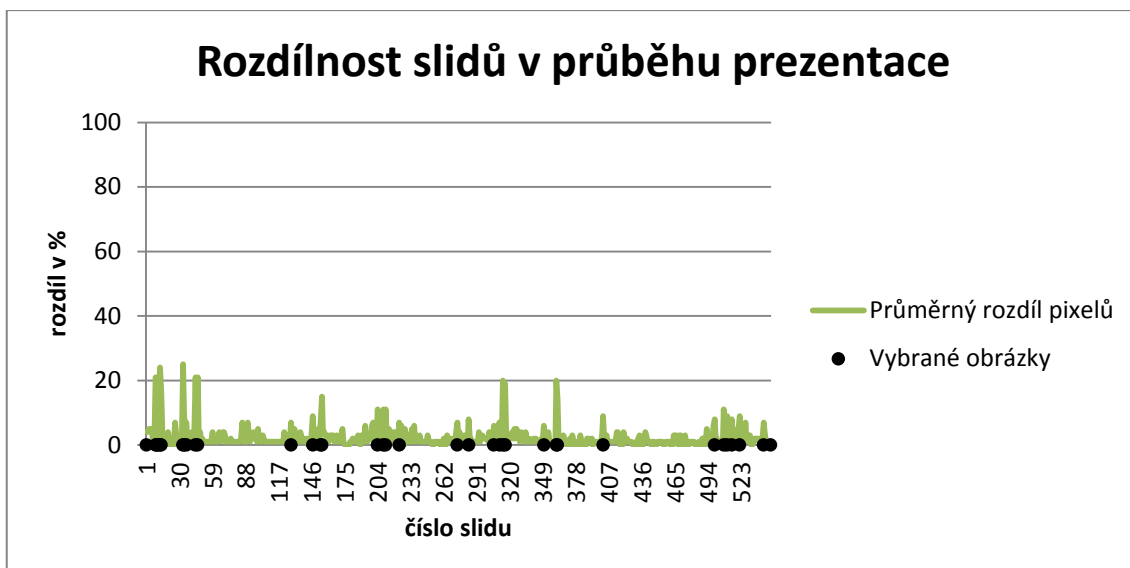
7.8.2 Test PIT – prezentace IT předmětů

Pro test přednášky typu PIT byla vybrána přednáška předmětu Grafické a databázové aplikace z akademického roku 2009/2010. Tato přednáška obsahuje 549 slidů, ale není zatížena chybou jako v předchozím případě. Větší počet obrázků v prezentaci je způsoben častějším přepínáním mezi okny v počítači typicky při ukázce softwaru v IT předmětech. Z tohoto počtu obrázků je opět nutné na základě porovnávacích metod vhodně vybrat sadu výsledných slidů. Ke konečné redukci bude opět použita kombinační metoda s váhovými koeficienty 2:1:7, protože prezentace obsahuje více slidů než je nastavených 150.



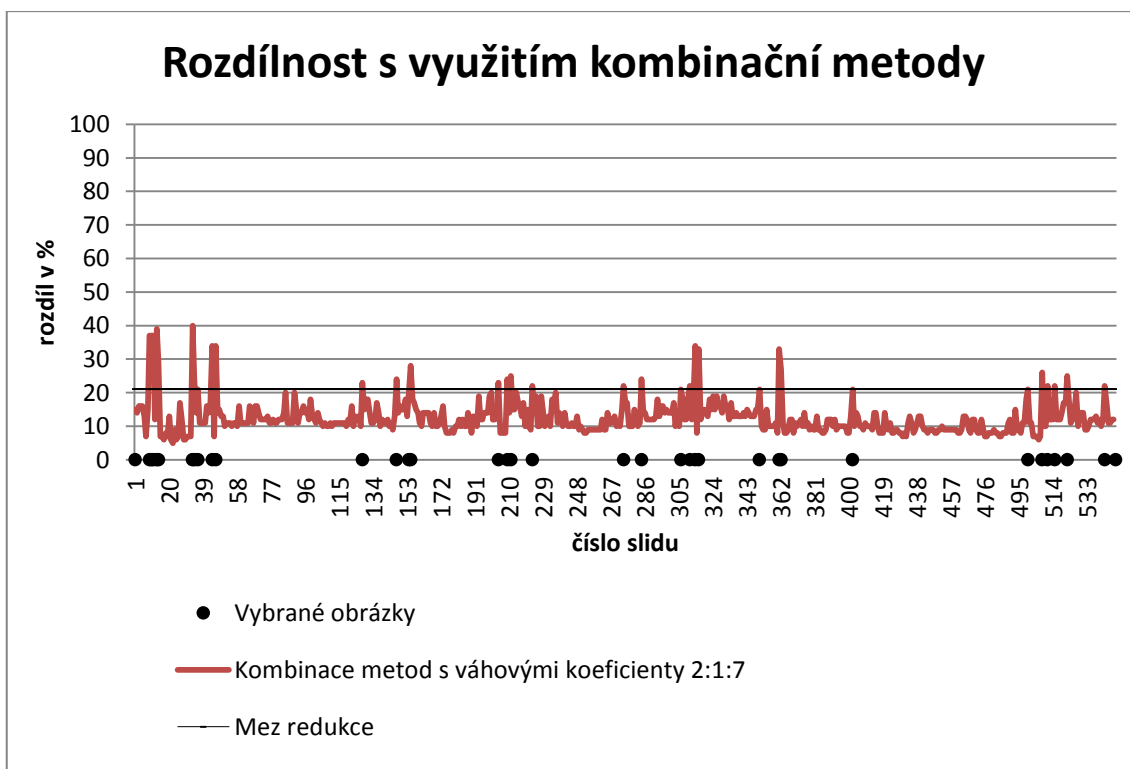
Graf 4: Rozdílnost slidů prezentace GDA 1. metodou

Graf 4 znázorňuje, že přednáška obsahuje mnoho rozdílných slidů. V tomto případě by se redukce dala provést přímo z 1. Metody tak, aby aplikace vybrala nastavený počet slidů v prezentaci.



Graf 5: Rozdílnost slidů prezentace GDA 3. metodou

Z grafu 5 je vidět, že tato metoda již dokáže na tomto typu prezentací vyloučit některé méně rozdílné slidy a tedy funguje jako jakési jemnější síto pro výběr obrázků do vybrané sady. V konečném důsledku je opětovné použití kombinační metody v poměru 2:1:7 výhodné. Z předešlého grafu již jsou patrná místa v prezentaci, ve kterých se významné slidy budou nacházet.



Graf 6: Rozdílnost slidů prezentace GDA kombinační metodou

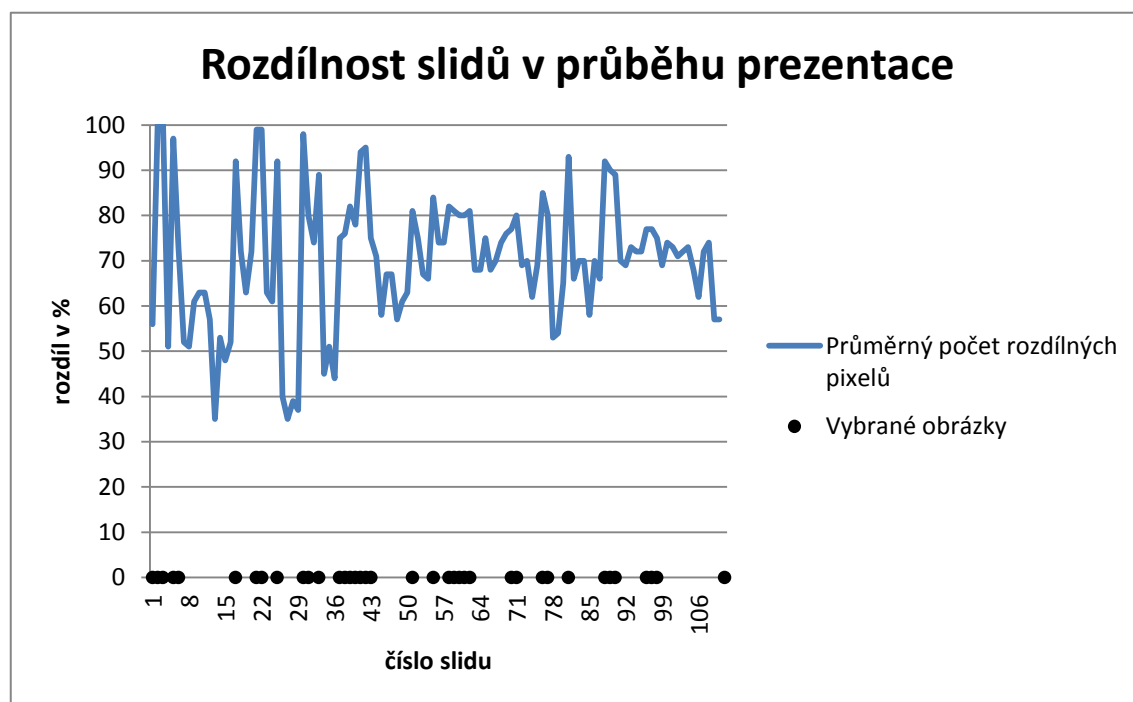
V grafu kombinační metody jsou již velmi dobře vidět místa, ve kterých se nacházejí slidy, které se dostanou do vybrané sady obrázků. Lze tedy soudit, že program

dokáže automaticky zpracovat taktéž prezentace, ve kterých se pracuje jak s klasickými výukovými slidy, tak s ukázkami softwaru a dalších nástrojů v počítači. Prezentace tohoto typu jsou nejčastějším typem prezentací, které se momentálně v učebně vybavené systémem Mediasite zaznamenávají. U této konkrétní prezentace bylo z původního počtu 549 slidů vybráno 36 obrázků.

7.8.3 Test PVP – přednáška s výkladem pojmů

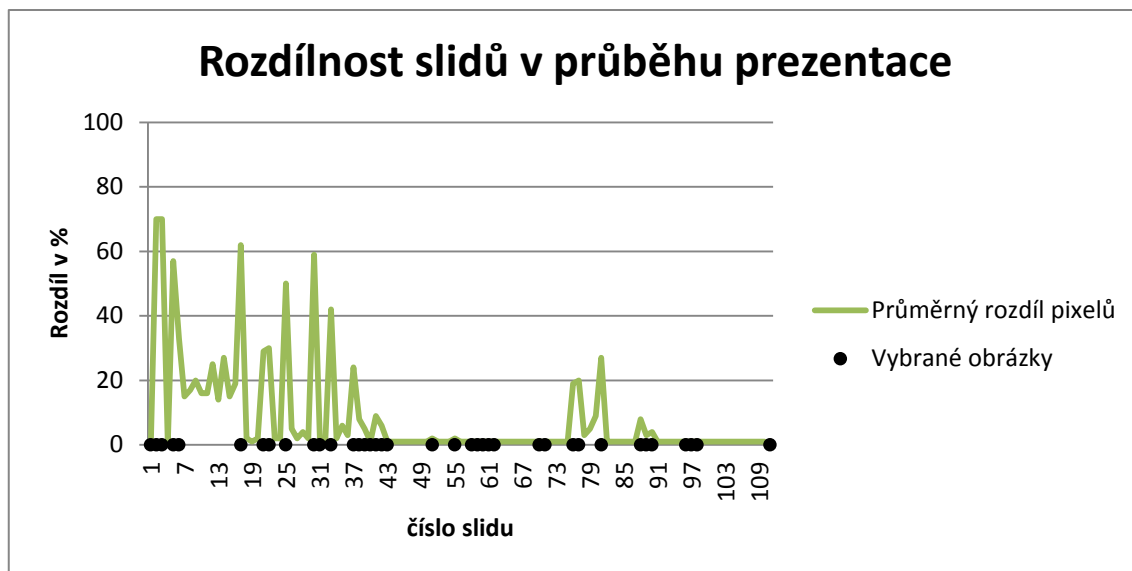
Pro tento typ byla zvolena přednáška Projektování automatizovaných systémů z akademického roku 2009/2010. U tohoto druhu přednášek se velmi často využívá vizualizér pro snímání plochy na stole vyučujícího. Toto zařízení pořizuje v určitých intervalech snímky toho, co se na určené ploše nachází. Vyučující tímto způsobem buď píše na papír poznámky pro studenty, nebo může ukazovat jakékoliv věcné předměty. Tím, že zařízení snímá v předem daných intervalech, dochází opět k vygenerování mnoha redundantních slidů. V tomto případě je situace ještě obtížnější, protože snímací zařízení ne zcela přesně zaostřuje na zmíněnou plochu a tím je každý obrázek trochu jiný i když je snímán stále stejný objekt.

Protože přednáška původně obsahovala 111 slidů, budou pro kombinační metodu zvoleny váhové koeficienty v poměru 6:1:3.



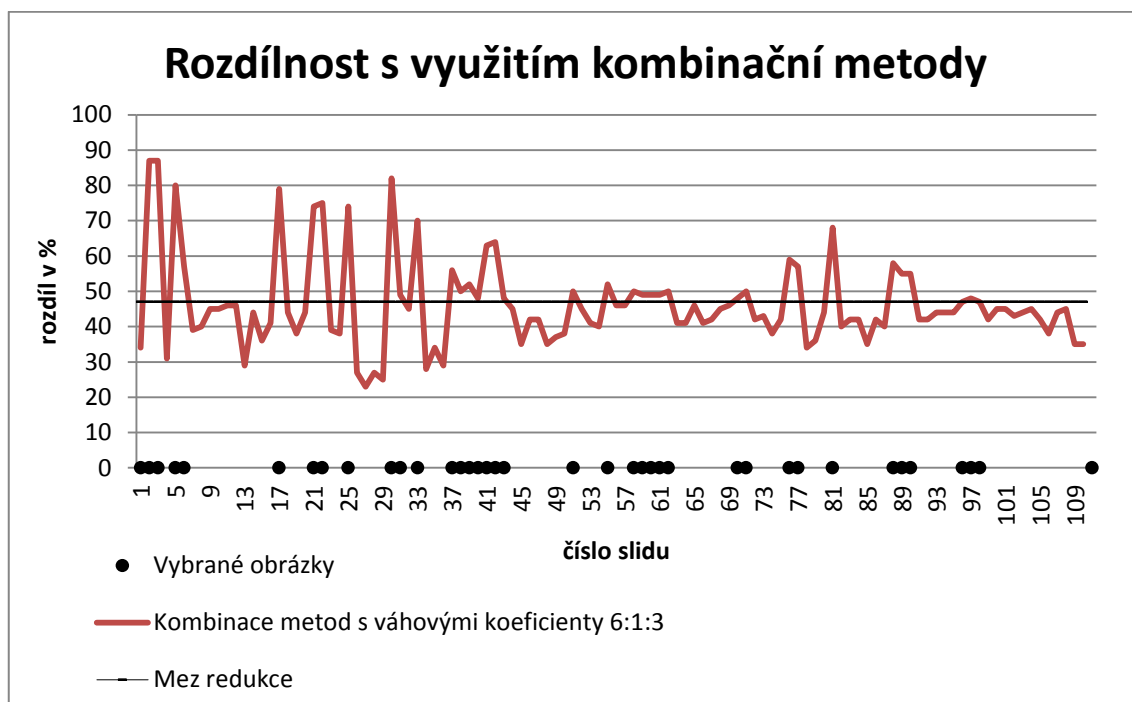
Graf 7: Rozdílnost slidů prezentace PAS 1. metodou

Graf 7 zobrazuje metodu průměrného počtu rozdílných pixelů, jejíž váha je zvolena jako nejvyšší. Již z tohoto grafu je znatelný výskyt klíčových momentů prezentace. Z grafu dále vyplývá i tvrzení, že se jednotlivé snímky od sebe poměrně značně liší i přes to, že je na nich stále zobrazen stejný obsah.



Graf 8: Rozdílnost slidů prezentace PAS 3. metodou

V tomto případě slouží metoda průměrného rozdílu pixelů opět pro větší zjemnění rozdílu prezentace, což je na grafu 8 dobře vidět v místě kde byli vybrané slidy, ale rozdíly této metody zde nejsou téměř žádné. Tento jev je způsoben vyšší vahou 1. metody porovnávání.



Graf 9: Rozdílnost slidů prezentace PAS kombinační metodou

Z grafu kombinační metody jsou již znatelné extrémy v rozdílech v prezentaci. Tyto extrémy jsou nakonec vybrány do konečné vybrané sady. Z původních 111 snímků je po úpravě v navigaci pouze 38 snímků, kterými lze mnohem jednodušším způsobem přednášku procházet.

Z výsledků lze usoudit, že aplikace je připravena i na tento typ prezentace, kdy dochází k větší redundanci obrázků a počet slidů není tak vysoký jako u jiných prezentací. Rozložení vybraných slidů tedy lépe vystihne průběh prezentace, aby student zachytil důležitý moment a mohl se k němu případně vrátit.

Předchozí testy naznačují, jakým způsobem aplikace vybírá důležité body v prezentaci. Aplikace je připravena na automatický post-processing přednášek s velkým počtem slidů, který může přesáhnout řád tisíce, ale zároveň zajišťuje zpracování méně rozsáhlých prezentací, u kterých je nutné navigaci pouze nepatrně zkorigovat.

Závěr

Diplomová práce je příspěvkem k optimální implementaci záznamů přednášek ve speciálním formátu richmédii do e-learningového systému na Fakultě mechatroniky, informatiky a mezioborových studií. V části věnované systému Mediasite bylo popsáno řešení, které se v současnosti na Technické univerzitě v Liberci používá k záznamu velkého množství přednášek. Byl nastíněn problém ve výsledné webové prezentaci záznamu přednášek. V důsledku chyb snímání nebo využití doplňkového snímacího zařízení vzniká velký počet redundantních slidů a je ztracena výhoda digitální navigace pomocí slidů. Diplomová práce tento problém řeší a nabízí editaci takto poškozené prezentace v automatickém nebo poloautomatickém režimu.

V rámci řešení implementace systému Mediasite byla vyvinuta aplikace, která byla nazvána RichMedia Slider, která daný problém řeší. Aplikaci lze přidat do fáze post-processingu před vlastní publikací prezentací zaznamenaných přednášek. V průběhu vývoje vzniklo několik zásadních problémů, které bylo nutné vyřešit. Hlavním úskalím, které se povedlo úspěšně překonat, byla práce s operační pamětí počítače. Ve chvíli, kdy bylo potřebné načíst řádově tisíce slidů, byla paměťová náročnost příliš vysoká, a tedy bylo nutné navrhnout optimálnější řešení. Vyřešit bylo potřeba také problém jakým způsobem všechny načtené slidy zobrazovat a jak s nimi vhodně uživatelsky pracovat. Dále bylo nutné optimalizovat rozhraní aplikace, aby ho uživatel mohl intuitivně ovládat, ale aby většina potřebných činností byla provedena automaticky. Problém byl také v tom, jakým způsobem vhodně zkombinovat předem navržené porovnávací metody a případně v jakém poměru, aby tuto část post-processingu bylo možno co nejvíce automatizovat. Z tohoto důvodu byla provedena řada testů na několika prezentacích, které na danou otázku daly odpověď. Největším úspěchem bylo vyřešení celkové úpravy HTML šablony Mediasite. Šablona disponuje několika skripty v jazyce *JavaScript*, které bylo nutné upravit do takové míry, aby se oddělila logika nasnímané plné sady obrázků od logiky redukované sady vycházející z vyvíjené aplikace.

Z hlediska testování se úspěšně povedlo ověřit funkčnost aplikace RichMedia Slider. Testování se provádělo na typech prezentace PIT, PVP a PP podle [3]. Nejrozšířenějším typem je v tuto chvíli prezentace PIT – přednášky IT předmětů. Proto

se celý vývoj prováděl hlavně na prezentaci tohoto typu. Testy prokázaly, že aplikace úspěšně zpracuje i ostatní typy prezentací.

Ve spojení s testováním vzniklo několik návrhů, které se samotného systému Mediasite týkají, a které by do budoucna významným způsobem odlehčily post-processingu. Zejména v prezentacích typu PVP, kdy jsou prezentace doplněny výkladem a ukázkou obsahu z knih nebo ukázkou předmětů skrze snímací zařízení, by bylo vhodné doplnit ovládací prvky celého systému v učebně o tlačítko „sejmi plochu“, které by na daný impulz pořídilo snímek plochy, na které se předmět, kniha nebo text nachází.

Další návrh se týká způsobu stahování přednášek ze serveru Mediasite do počítače pro post-processing, který se momentálně provádí převážně v nočních hodinách. Pokud by se do systému úspěšně implementovala funkčnost, která by přednášky stahovala v mezech mezi předměty a v čase, kdy je systém neaktivní, bylo by možné pouze spustit aplikaci vycházející z této práce a vygenerovat upravenou šablonu v kratším čase, než se tomu děje dnes. Celkový čas post-processingu by se díky této úpravě snížil z několika hodin na desítky minut.

Aplikace, která v rámci diplomové práce vznikla, umožňuje úspěšně redukovat počet snímků prezentace na přijatelnou hodnotu. Díky rozsáhlosti zdrojových kódů systému Mediasite se dá očekávat pokračování vývoje programu RichMedia Slider za účelem optimalizace zdrojových kódů HTML šablony.

Literatura

- [1] *About OCR* [online]. 2011 [cit. 2011-05-11]. ABBYY Fine Reader. Dostupné z WWW: <http://finereader.abbyy.com/about_ocr/>.
- [2] BUREŠ, Miroslav. *Nová generace webových technologií*. 1. Praha : Vox, 2005. 264 s. ISBN 80-86324-46-X.
- [3] CÍSAŘOVÁ K., KOPETSCHKE I., HNÍDEK J.: Inteligentní multimediální e-learningový portál – Prostor pro pokročilé vzdělávání, MoodleMoot.cz 2010, Brno
- [4] ČERNÝ, Michal. Vyhledávejte v multimédiích. CHIP. 2010, 05, s. 32-33.
- [5] *Grafika.cz* [online]. Institut digitální fotografie, 2008, 11.5.2011 [cit. 2011-05-11]. Grafika on-line. Dostupné z WWW: <www.grafika.cz>.
- [6] Image processing in Delphi [online]. 2007 [cit. 2011-05-11]. Image processing in Delphi. Dostupné z WWW: <<http://imageprocessingindelphi.blogspot.com/>>.
- [7] KOSEK, Jiří. *HTML tvorba dokonalých WWW stránek*. Praha : Grada, 2004. 296 s.
- [8] MURRAY, James D.; VANRYPER, William. *ENCYKLOPEDIE GRAFICKÝCH FORMÁTŮ*. 1. Brno : Computer Press, 1995. 736 s. ISBN 80-85896-18-4.
- [9] PLÍVA, Zdeněk a Jindra DRÁBKOVÁ. 2009. Metodika zpracování diplomových, bakalářských a vědeckých prací na FM TUL. Liberec : TUL. ISBN 978-80-7372-189-3. Dostupné též z: https://www.ite.tul.cz/ite/images/data/edu/jak_psat_dp.pdf.
- [10] SKÁLA, Václav. *Algoritmy počítačové grafiky I. - III.*. 1. Plzeň : Západočeská univerzita, 1992.
- [11] Sonicfoundry [online]. 2010 [cit. 2011-05-11]. Mediasite. Dostupné z WWW: <<http://www.sonicfoundry.com/Mediasite/>>.
- [12] SVOBODA, Luděk, et al. 1001 tipů a triků pro Delphi. 2. aktualiz. vyd. Praha : Computer Press, 2003. 552 s. ISBN 80-722-6488-5.
- [13] TRÍSKOVÁ, Lenka. *GNU nástroje pro tvorbu WWW stránek*. Praha : Grada, 2005. 244 s. ISBN 80-7169-861-X.
- [14] Umíme to s Delphi. Živě. 2001-2005. Brno : CPress Media, 2001-2005. Dostupný z WWW: <www.zive.cz/Clanky/Umime-to-s-Delphi--1-dil/sc-3-a-30959/default.aspx>. ISSN 1212-8554.

- [15] ŽÁRA, Jiří, et al. Moderní počítačová grafika : Kompletní průvodce metodami 2D a 3D grafiky. 2. Brno : Computer Press, 2005. 612 s. ISBN 80-251-0454-0.

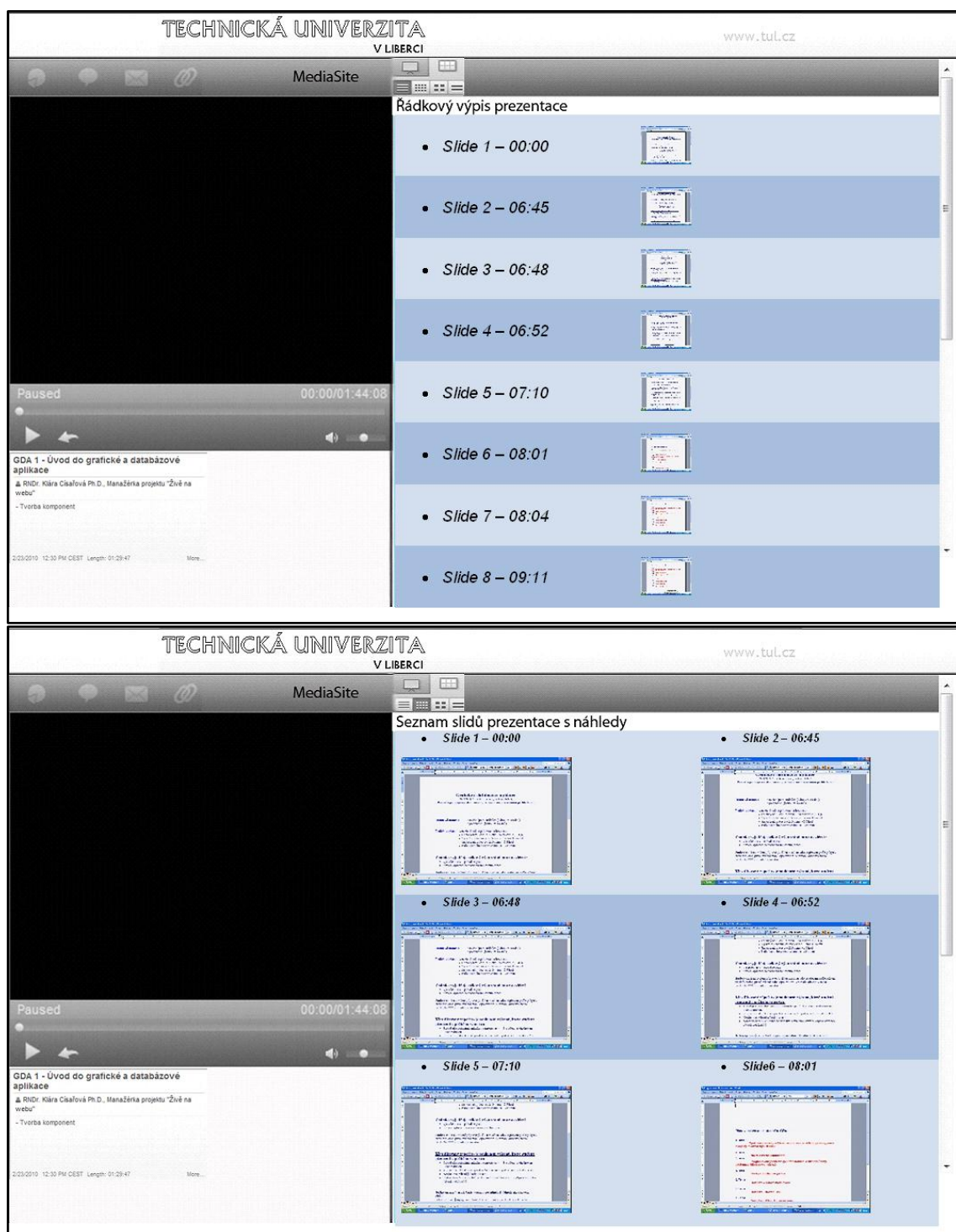
Příloha A – Popis přiloženého CD

Na přiloženém CD je k dispozici kompletní zdrojový kód aplikace RichMedia Slider spolu s jeho instalátorem. Kromě této diplomové práce v elektronické podobě lze na CD také nalézt výsledky testování deseti typových prezentací spolu s jejich zpracovanými grafy, které vyjadřují závislosti mezi porovnávacími metodami a které posloužily k určení doporučených hodnot do aplikace.

Na disku je k dispozici následující struktura adresářů:

- /Diplomová práce/
Vysázený text této diplomové práce
- /Instalátor/
Obsahuje instalační soubor pro aplikaci RichMedia Slider
- /Výsledky testů/
Obsahuje sešity v aplikaci Microsoft Excel s výsledky testů na typových prezentacích
- /Zdrojové kódy/
Zdrojový kód pro aplikaci RichMedia Slider, které je možné zkompileovat v Borland Delphi 7. Obsahuje také zdrojové kódy nápovědy, které lze zkompileovat v prostředí HelpMaker

Příloha B – Návrhové zobrazení optimalizované navigace



TECHNICKÁ UNIVERZITA
V LIBERCI

www.tul.cz

MediaSite

Rádkový výpis klíčových bodů prezentace

- Slide 1 – 00:00
- Slide 2 – 09:54
- Slide 3 – 09:56
- Slide 4 – 10:02
- Slide 5 – 10:12
- Slide 6 – 10:13
- Slide 7 – 16:20
- Slide 8 – 18:15

Paused 00:00/01:44:08

GDA 1 - Úvod do grafické a databázové aplikace
 a RNDr. Klára Čislová Ph.D., Manažerka projektu "Živě na webu"
 - Tvorba komponent

2/23/2010 12:30 PM CEST Length: 01:29:47 More...

TECHNICKÁ UNIVERZITA
V LIBERCI

www.tul.cz

MediaSite

Seznam klíčových bodů prezentace

• Slide 1 – 00:00

• Slide 2 – 09:54

• Slide 3 – 09:56

• Slide 4 – 10:02

• Slide 5 – 10:12

• Slide 6 – 10:13

Paused 00:00/01:44:08

GDA 1 - Úvod do grafické a databázové aplikace
 a RNDr. Klára Čislová Ph.D., Manažerka projektu "Živě na webu"
 - Tvorba komponent

2/23/2010 12:30 PM CEST Length: 01:29:47 More...